1

LEVEL II

AD-A062809

R-1136

A NEW BASELINE FOR THE INERTIAL NAVIGATION STRAPDOWN SIMULATOR PROGRAM

VOLUME III Program Description and Users Guide

by

R.J. Nurse, J.T. Prohaska, D.G. Riegsecker

July 1978

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

78 ... 24

Best Available Copy

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER R-1136	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) A New Baseline for the Inertial Navigation Strapdown Simulator Volumes I, II, III, and IV	5. TYPE OF REPORT & PERIOD COVERED 6/1/77 - 7/15/78
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. Nurse, J. Prohaska, D. Riegsecker	8. CONTRACT OR GRANT NUMBER(s) F33615-75-C-1149
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Charles Stark Draper Laboratory, Inc. 555 Technology Sq., Cambridge, MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project 6095, Task 4.2.5
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Avionics Laboratory Wright-Patterson AFB, Dayton, Ohio 45433	12. REPORT DATE July 1978
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.

ACCESSION for		
NTIS	White Section	X
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Inertial Navigation
Strapdown
Simulation
Instrument Modeling
Laser Gyros
Random Vibration
Single degree of freedom gyros and accelerometers

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This four-volume report describes an updated and expanded version of a direct, digital, modular simulation of a strapdown inertial navigation system employing a wander-azimuth computational frame, and subject to a six degree of freedom random vibration environment. The original version of this simulation was developed under Task 4.2.3(a) of the above contract during 1975 and 1976.

(CONTINUED ON REVERSE)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The user may simulate not only the gross dynamics of the flight profile (from an external or internal profile generation) but also the angular and linear random vibrations resulting from gusts and turbulence acting on the airframe. The total environment is applied to the models of the inertial components (laser or SDR gyros and pendulous accelerometers). The resulting outputs of simulated IMU are summed in an interface module and compensated and scaled in the simulated navigation computer. The latter also contains the velocity/attitude algorithm, which computes the body-to-inertial transformation, using either the direction cosine matrix or quaternion, and the navigation algorithm which numerically integrates the specific forces after transformation to the local vertical, wander azimuth computational frame. The outputs of the simulated navigation computer are the computed position, velocity, and attitude of the vehicle with respect to a local vertical, north pointing frame. The flight profile and the differences between it and the simulated navigation computer outputs are tabulated in an evaluation module for printing, plotting, or post processing.

A ground alignment Kalman filter for the INSS, also developed under this task, is not documented in this report, but may be available from AFAL/RWA-2 or -3.

The program is written in Fortran IV for use on a CD6600/CYBER74.

The report is structured as follows:

- Volume I is the Introduction and Summary
- Volume II contains analytical development of the equations to be mechanized and the transition to difference equation form
- Volume III is the Program Description and User's Guide
- Volume IV contains Program Listings.

R-1136

A NEW BASELINE FOR THE INERTIAL NAVIGATION
STRAPDOWN SIMULATOR PROGRAM

VOLUME III Program Description and Users Guide

by

R.J. Nurse, J.T. Prohaska, D.G. Riegsecker

July 1978

Approved: W. Denhard
W. Denhard

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

ACKNOWLEDGEMENT

This four-volume report was prepared under USAF Contract No. F33615-75-C-1149 by the Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts, in accordance with Section 4 of the contract. The monitoring Air Force project engineer is Mr. Neil Banke (RWA-2), Air Force Avionics Laboratory, Dayton, Ohio.

The Draper Laboratory program manager for this task is John Harper and the lead engineer is Roy Nurse. The authors of the report are Roy Nurse, John Prohaska, and Darold Riegsecker.

The authors express their appreciation to Messrs. William Shephard, David Kaiser, Stanton Musick, and Maj. Robert Edwards of the Air Force Avionics Laboratory and William Ostanek, Jane Goode, Tom Thorvaldsen, and William Caffery of the C. S. Draper Laboratory for their assistance during the course of this contract.

VOLUME III

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

APPENDICES

VOLUME III

LIST OF ILLUSTRATIONS

NOTE:
(Each Module Flow Diagram is identified as Figure 1 within the Module Description it pertains to.)

VOLUME III

LIST OF TABLES

TABLE PAGE

SECTION 1

INTRODUCTION

1.1 Purpose

This volume provides a detailed description of the simulator program from a functional, structural, and operational point of view.

The primary purpose of the program is to permit the evaluation of the navigation and attitude errors resulting from the mechanization of a strapdown inertial navigation system, using detailed sensor models, in a highly dynamic tactical aircraft environment. This purpose should be complementary to that of another CSDL-developed simulation (R-977), which assumed "perfect" sensors, and was directed towards the evaluation of the computational errors resulting from the mechanization of the navigation and attitude equations in a digital computer. During the latter part of the present task the simulated navigation computer software was modified to conform with the "upgraded" version of the algorithms employed in the reference insofar as was possible without changing the structure of the program.

The equations developed or presented in analytical form in Volume II, appear in this volume in program mnemonics in "Fortranesque" form.

1.2 Program Function

Although the program consists of a number of quasi-independent, replaceable modules, as will be described later, it may be more easily understood in terms of the four major functions performed.

The first function is the aircraft profile or trajectory generation. The intended source of the gross vehicle dynamics (position, $\underline{p}$) velocity, $\underline{v}$; attitude, $\underline{\phi}$, specific force, $\underline{a}^b$; angular velocity and acceleration, $\underline{\omega}^b$ and $\dot{\underline{\omega}}^b$) is the AFAL profile generator program, PROFGEN. For test and checkout, the present version of the INSS employs a simplified, self-contained, non maneuvering, "gross" profile generator. The linear and angular vibration environment is generated in another module and superimposed on the gross trajectory.

The profile generator provides both a reference profile and the forcing functions which drive the simulated sensors.

The second function is the simulation of the dynamics of the inertial measurement unit (IMU) and the barometric altimeter. Three single degree of freedom, floated, rate integrating, wheel-type gyros or three ring laser gyros may be simulated, plus three, single-axis pendulous accelerometers. Specific force and angular velocity from the profile or trajectory are the primary forcing functions for the inertial components, while the models are defined by the input parameters for the sensors.

The IMU simulator provides the incremental angles, $\underline{\Delta\theta}^g$, and incremental velocities, $\underline{\Delta V}^a$, and the indicated barometric altitude, h_B, to the navigation computer interface unit, just as if they were obtained from actual sensors.

The third function is the simulation of a navigation (and attitude) computer for a strapdown aircraft inertial navigation system (INS). The first major subtask of the navigation computer is to correct the $\underline{\Delta V}^a$'s and $\underline{\Delta\theta}^g$'s, using the loaded values of the component parameters and the compensation models, so that the resultant computed values, $\underline{\Delta V}_c^b$ and $\underline{\Delta\theta}_c^b$, approach the integrals of the specific force and angular rate, output by the profile generator.

The second major subtask of the navigation computer is the updating of the body to inertial transformation, C_b^i, using the $\underline{\Delta\theta}_c^b$'s, and the subsequent transformation of the $\underline{\Delta V}_c^b$'s from the body to the inertial frame.

The final major subtask of the navigation computer is the updating of the position, $\underline{p}_c$ and velocity, $\underline{V}_c$, and the extraction of the attitude angles, $\underline{\phi}_c$, in the navigation and attitude algorithms.

The fourth function is the evaluation of the errors produced by the second and third functions relative to the reference values generated by the profile.

The evaluation module outputs both the instantaneous, reference values of position, velocity, and attitude and the "errors" in the computed values of the same.

This gross functional breakdown of the INSS is illustrated in Figure 1-1.

1.3 Overview of Functional Structure

The INSS program consists of a number of modules, each of which performs part or all of one of the four major functions described in the preceding subsection.

Each module contains its own initialization data (IFILE) which includes both module parameter and timing and control. In addition, each has access to a physical data file (PDATA), which provides some physical constants and the initial conditions for the simulated flight.

There really is no executive for the INSS program in the normally understood sense, only a sequencer, which calls the modules in order and, in some cases, buffers some of the module input/output data. Hence in expanding the description of the program, the sequencer does not enter into the picture directly, beyond establishing the order in which the modules are called.

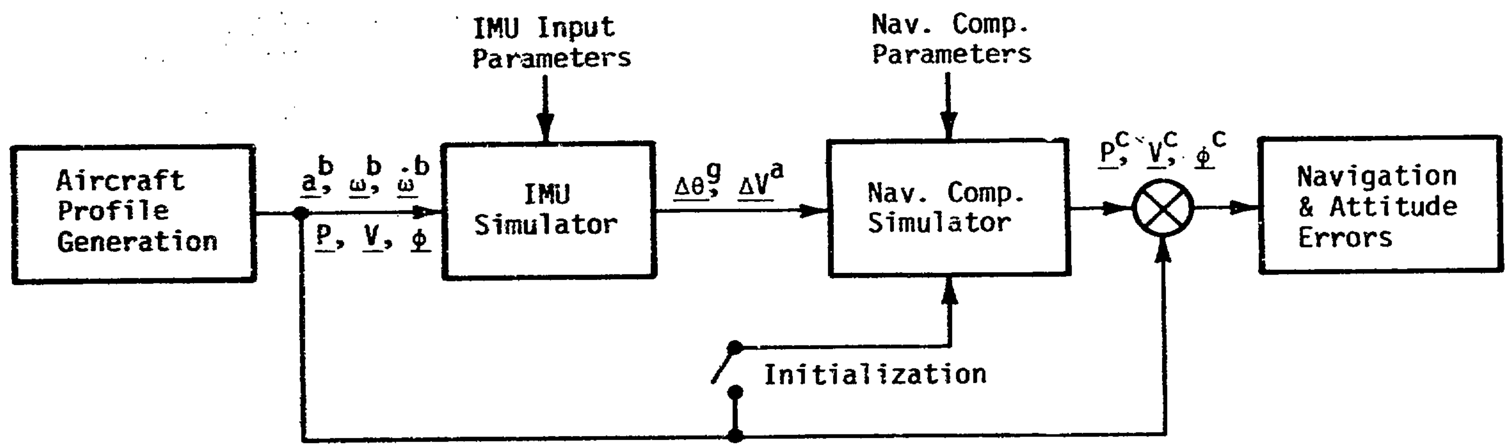

Figure 1-1. INSS gross functional diagram.

In terms of the individual modules the INSS program performs the following functions:

1. Provides a flight profile for a point-mass aircraft executing coordinated maneuvers. Profile includes position velocity, acceleration and attitude of the aircraft.

2. Superimposes on the flight profile, the angular and linear motion of the aircraft in response to random aerodynamic forces such as gusts and turbulence.

3. Operates on the specific force and angular velocity to produce the ideal body frame values of these quantities and the corresponding angular acceleration.

4. Integrates the gyro and accelerometer equations of motion, of the form specified, after accounting for the displacements of the inertial components from the "center" of the point-mass vehicle, and their orientations with respect to the body frame. Incorporates effects of component parameters and environment on component outputs.

5. Reads true altitude and perturbs the same according to the altimeter model.

6. Transmits sensor data from simulated IMU to simulated navigation computer, and resets sums.

7. Performs accelerometer compensation function (this is essentially the inverse of the accelerometer model - with errors and omissions).

8. Using compensated accelerometer outputs (ΔV's) and gyro outputs ($\Delta\theta$'s) performs the gyro compensation (again, this is essentially the inverse of the gyro model - with errors and omissions).

9. Updates the body to inertial frame transformation (direction cosine matrix, d.c.m.) and transforms the incremental velocities to the inertial frame (either a d.c.m. or a quaternion update may be employed).

10. Transforms the incremental velocities to the local vertical wander azimuth computational frame and computes local vertical position, velocity, and attitude (incorporating the barometric altimeter for vertical damping).

11. As required, the navigator outputs are differenced with the flight profile values and the resulting errors are printed out and/or plotted.

The foregoing description of the program flow is presented pictorially in Figure 1-2, where the dashed lines indicate the major functions shown in Figure 1-1.

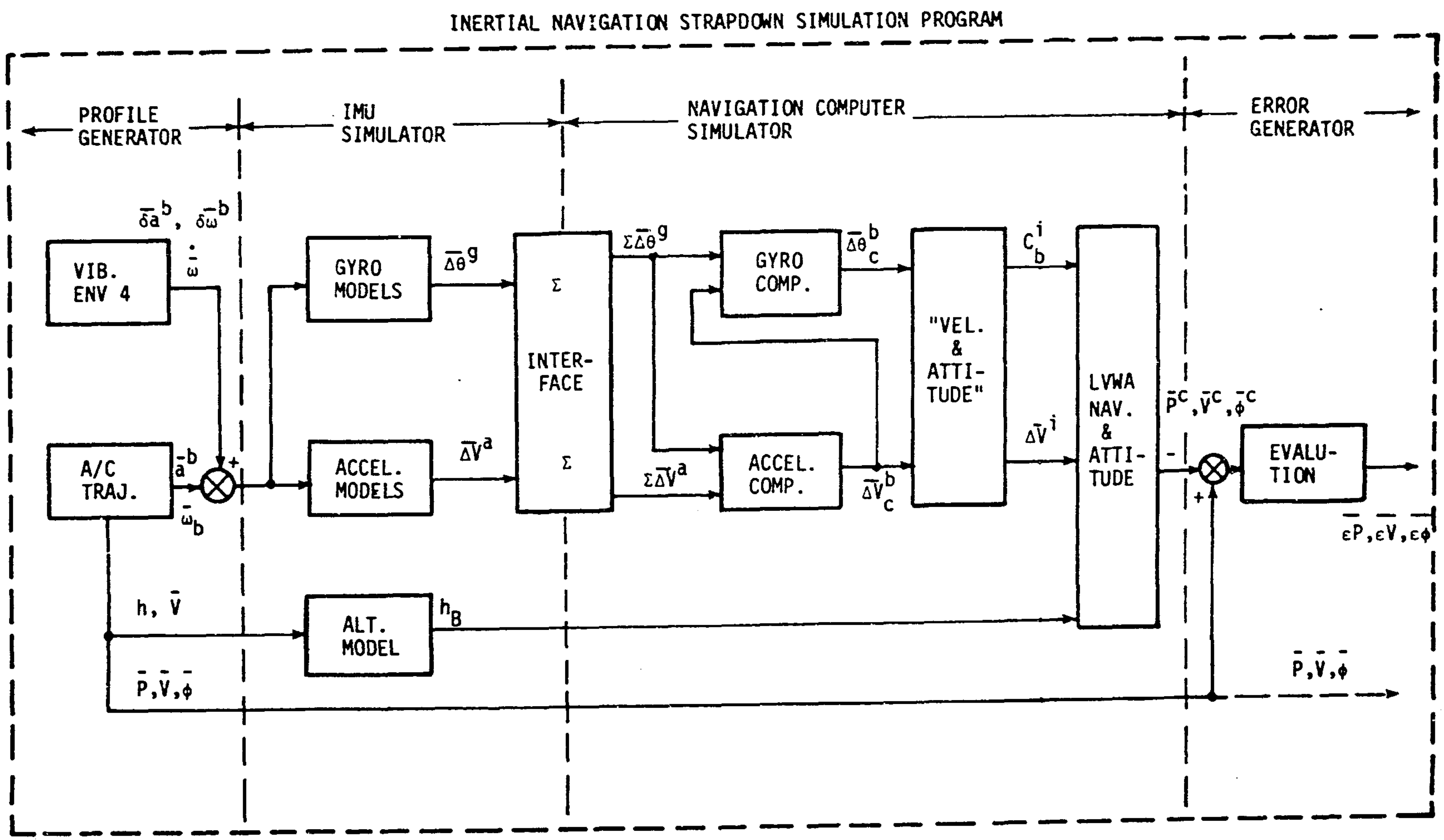

Figure 1-2 INSS Functional Block Diagram

SECTION 2

PROGRAM DESCRIPTION

2.1 Introduction

The INSS simulator consists of 11 subprograms (modules) which simulate the various hardware and software functions in a strapdown system. A sequencer (main) program interrogates each module in turn at one of two different user-specified frequencies. Associated with each module are initialization data files that also contain user-specified switches and timing specifications so that each module interrogated by the sequencer can control its own operating time, data output frequencies, initial- and last-pass program functions. Each module initialization data file (IFILE) has a unique input/output (I/O) file unit number which is identified with each module. Also, every module has access to a common data file (PDATA) that contains physical data and initial-trajectory flight-path parameters required to initialize certain modules.

In Table 1 are listed the INSS program modules. The default initialization data files are listed by name and FORTRAN unit number in Table 2. Two replacement modules are listed with their default data file names in Table 3.

In addition there is a set of library subroutines for the purpose of performing repeated common calculations.

- o Matrix transpose by matrix multiplication (MTXM)
- o Matrix transpose by vector multiplication (MTXV)
- o Matrix by matrix multiplication (MXM)
- o Matrix by vector multiplication (MXV)
- o Gaussian random number generator with specifiable mean and standard deviation (GAUSS)

Table 1. INSS Programs

Program Dataset	Subroutine	Function
INSSEQ.FORT	Main	Executive
INSTRJ.FORT	TRJ	Interim Trajectory Data-Processor/ Interface Module
INSENV.FORT	ENV	Hardware Vibration Simulator Module
INSGYR.FORT	GYROS	Hardware Simulator Gyro Module
INSACC.FORT	ACCEL	Hardware Simulator Accelerometer Module
INSALT.FORT	ALTI	Hardware Simulator Altimeter Module
INSRDR.FORT	RDR	Hardware/Software Interface Module
INSCAC.FORT	ACOMP	Software Accelerometer-Compensation Module
INSCGY.FORT	GCOMP	Software Gyro-Compensation Module
INSALG.FORT	ALG	Software Velocity Attitude Module
INSLLN.FORT	LLN	Software Navigation Module
INSEVL.FORT	EVL	Evaluation Module

Table 2. Data Files

Initialization Data Base	I/O File Unit No.	Initialization Data Base	I/O File Unit No.
SEQ.DATA	10	RDRD.DATA	65
TRJD.DATA	20	CACD.DATA	67
ENVD.DATA	30	CGYD.DATA	69
GYRD.DATA	40	ALGD.DATA	70
ACCD.DATA	50	LLND.DATA	80
ALTD.DATA	60	EVLD.DATA	90
		PHYSD.DATA	7

Table 3. Replacement modules and corresponding data sets.

Program Dataset	Sub-routine	Initialization Dataset	I/O File Unit No.	Function
INSGYRL.FORT	GYROS	GYRLD.DATA	40	Hardware Simulator Laser-Gyro Module
INSCGYL.FORT	GCOMP	CGYLD.DATA	69	Software Laser-Gyro Compensation Module

There is also a set of subroutines that are required solely by the local level navigator module (LLN). These subroutines have the following functions,

- o Angular velocity calculation (ANGVEL)
- o Angular velocity by DT and Coriolis correction (TORCOR)
- o DCM second order update matrix (AUP)
- o Gravity computation (GRAV)
- o 3x3 matrix multiply (MM)

The INSS simulator has the operational flexibility to allow for the replacement of modules having the same function but different formulation. The initialization files for each of the modules may be readily replaced in total or particular data elements within the data file may be replaced. Printed output frequency may be controlled by the selection of a printing interval variable in the initialization file of the modules.

2.2 Operation

(1) Module Replacement

In order to replace a module having the same function but different formulation, it is only necessary to retain the same subroutine name and argument list. Module replacement is accomplished merely by physically substituting the appropriate module deck in a batch process computer run or by editing the program files in the computer storage. The actual replacement method is peculiar to the computer setup available.

(2) Data File Replacement

Each of the modules has an associated initialization data file. Again, for convenience, the whole data file may be replaced by a new file of data values. This operation may be the physical replacement of data deck in a batch process or substitution of a data file from a disk or tape storage. A new data file usually accompanies the substitution of a new program module.

(3) Data Value Override

Within any particular initialization file, it is often necessary to use a different data value for some variable. To accomplish this overriding for each variable, it is only necessary to append a new record at the end of the initialization file to be changed. This record requires an (15, F20.10) format for reading an index number, IX, and data value, DATA(IX). The index IX, referes to the variable number in the file, for which an override is desired and the variable DATA(IX) contains the value of the variable. Any number of overrides may be appended for the same or different variables within each of the initialization files, since only the latest value is retained.

(4) Simulation Termination

A simulation is normally terminated by the user specifying the termination time, TEND, in the sequencer (SEQ) initialization file. When the simulation time reaches the value TEND, a last pass indicator, IENDF, is set to 1, so that each module will receive a last pass to terminate normally.

(5) Data Output and Print Control

There are several modes for outputting data from the simulator. From each module, data may be output at a user specified frequency as described in the flow chart, Figure 2-1. The evaluation module (EVL) supplies the basic output of the simulator and an identical data file to the evaluation module print file may be stored on disc or tape.

To provide sufficient flexibility in simulator-data output and module print control, three user-specified data items are provided to

(1) Allow general data output of all modules at the same frequency.

PRNTDT (s) This data item must be specified in the INSS common data file (PDATA). This item provides a general data-output print frequency for all modules and overrides any values specified in MODPDT in each module

If PRNTDT=0, module printing occurs only if MODPDT>0 and PRNTSW=1. Printing will then occur at the frequency specified in MODPDT of each module. If MODPDT=0, no module printing occurs.

If PRNTDT≥0, module printing occurs for all modules at the frequency specified in PRNTDT if PRNTSW=1 regardless of the values of MODPDT for each module.

(2) Allow individual module printing at different frequencies.

MODPDT (s) This data item must be specified in the initialization data file of each module. This item is used to control the frequency of module output data printing and provides capability for different modules to print at different frequencies.

If MODPDT=0, this state is tested only if PRNTDT=0, and then no module printing occurs.

If MODPDT>0, module printing occurs at the frequency specified in MODPDT, provided PRNTDT=0 and PRNTSW=1.

(3) Provide an individual module-printing on/off switch to control printing in each module.

PRNTSW — This data item must be specified in the initialization data file of each module and is simply an on/off switch.

If PRNTSW=0, no module printing occurs regardless of the contents of MODPDT and PRNTDT.

If PRNTSW≥1, module output data printing occurs provided both MODPDT and PRNTDT are not 0.

The evaluation-module (EVL) data output is in tables containing trajectory attitude, position, velocity, and the corresponding navigation errors which are printed online every 50 EVL-module operating cycles if PRNTSW=1. The user controls EVL-module operating frequency in the usual way by specifying a value for DT in the initialization data file. See Fig. 1 for Flow Chart. A tape or disc file may be written on FORTRAN unit 12 with the same format and at the same frequency as the printed output file.

(6) Memory Storage Requirement

The total storage required for the INSS system on an Amdahl computer is approximately 125k bytes.

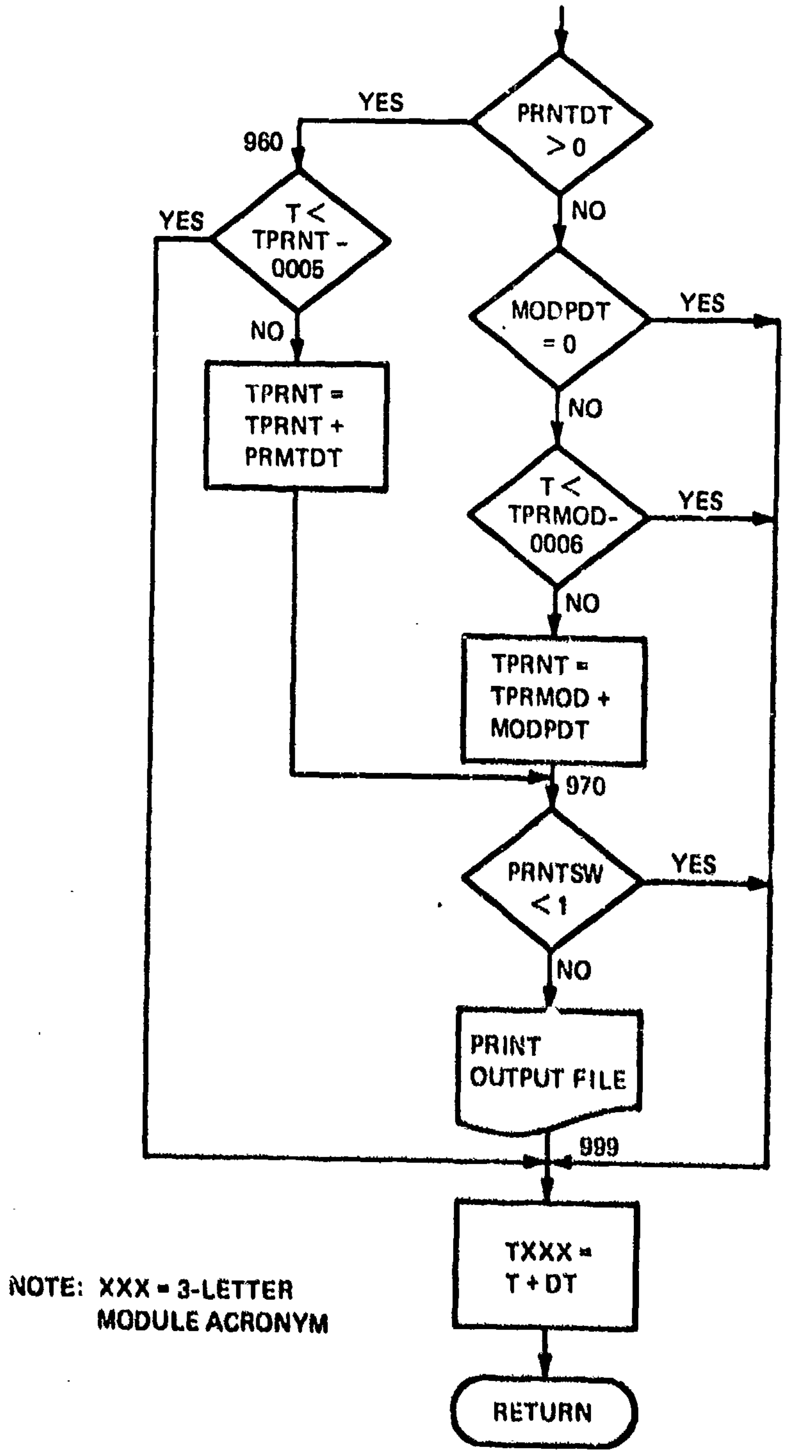

Figure 2.1 Data Output and Print Control

2.3 INSS Module Descriptions

The modules and subroutines available for use in the INSS simulator are described. Shown here is an index of the modules.

2.3.1 MAIN PROGRAM (SEQ) (INSS SEQUENCER)

(1) General Description

The SEQ module is the main program of the INSS. It functions as a synchronous executive by calling the hardware subprograms at a fast frequency and the software programs at a slow frequency. The two subroutine call frequencies reflect the different computation rates needed to simulate hardware and software. It should be further noted that each subprogram has its own operating frequency at which it performs its function. When called by SEQ, each subprogram performs its function anew only if its operating cycle time has elapsed. Otherwise it immediately returns control to SEQ. In order to synchronize the various operating frequencies the following restrictions apply to the various cycle times: the slow rate cycle time for calls to software subprograms must be an integer multiple of the fast cycle time for calls to the hardware subprograms; each subprogram operating cycle time must be equal to or a greater integer multiple of the cycle time at which it is called.

(2) Sequencer Module Flow Diagram

The general structure of the SEQ module logic, and the order in which all other modules are invoked are shown in Figure 1.* With regard to this flow chart, several points are worthy of particular attention. The slow cycle time, which is chosen to reflect computation frequency of the software functions in a strapdown navigator, is required to be an integer multiple of the fast cycle time that indicates the sampling frequency (or data availability) of the trajectory, vibration, and instrument measurements, exclusive of the altimeter module. The PDATA file is written on by the trajectory (TRJ) module only, and, with the exception of the evaluation (EVL) module, subsequently becomes an element common to all other modules.

---

*Each module flow diagram is identified as Figure 1 within the module description it pertains to.

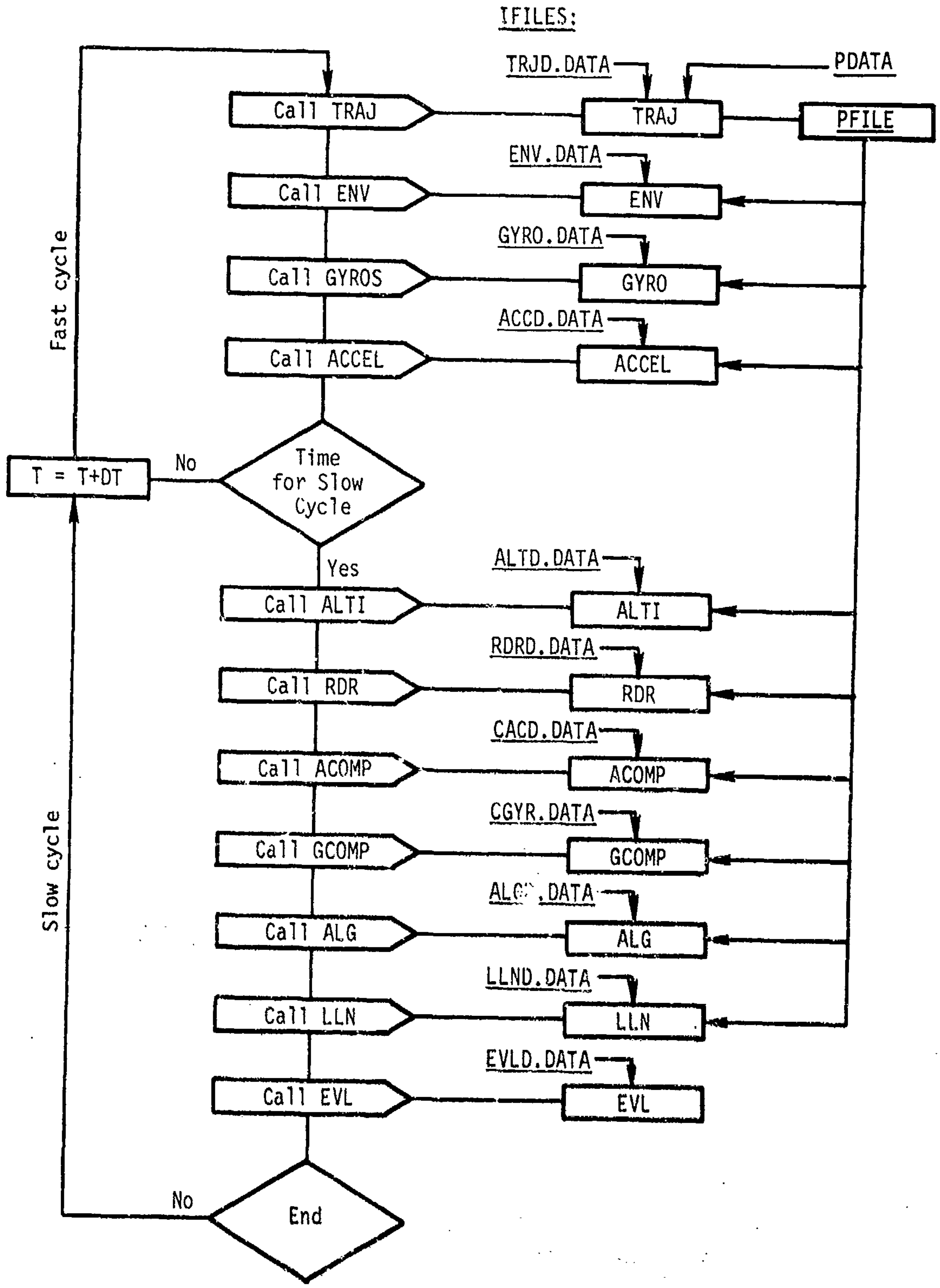

Figure 1. Sequencer module.

(3) Input

(a) Test Identification File

FORTRAN unit number: 5

FORTRAN format: 9A8

This file consists of one record containing a 72-character user-specified identification of a simulation run.

(b) Program Initialization File (IFILE)

FORTRAN unit number: 10

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.01	s	overall module operating cycle time
2	PRNTSW	1.0	logical	print switch 0 - no print otherwise - print output
3	OUTPSW	0.0	logical	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	-----	not used
6	SPARE2	0.0	-----	not used
7	TEND	60.0	s	simulation end time
8	DTSLOW	0.02	s	module slow operating cycle time
9	MODPDT	6.0	s	module print interval

(c) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.72921151470E-4	rad/s	earth rotation rate
2	RE	20975640.0	ft	earth radius
3	G	32.2	ft/s	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5 to 20	PBUF(1)	0.0	----	not used

(4) Call-Line Data

Call-line input-output data are not applicable to SEQ.

(5) Formulation

(a) Initialization

The following functions are performed during the first pass.

- SIMEND is initialized to 0.
- Read and print test identification.
- Read and print initialization data (IFILE).
- Read and print common initialization data (PFILE).
- Set OFILE = XFILE.
- Initialize simulation time at T = 0.
- Each of the program modules is called in sequence as illustrated in the General Functions Section.

(b) General Functions

SEQ is primarily a series of FORTRAN CALL statements which call each module in turn at the user-specified sequencer operating frequency. All interface data are passed from one module to another by parameter lists where dummy variables are used to protect input data.

The following is the series of CALL statements sequenced every operating cycle. The parameter lists are shown with the variable rather than the dummy name. Data items on the second line of each statement are module output.

CALL TRAJ

$$\overbrace{\text{(T, IENDF, PRNTSW, MODPDT,}}^{\text{INPUT}}$$

$$\underbrace{\text{LAT, LON, ALT, VEL, PITCH, YAW, ROLL, DVT, HDING, WANDER, AB, WB)}}_{\text{OUTPUT}}$$

CALL ENV

$$\overbrace{\text{(T, IENDF, AB, WB,}}^{\text{INPUT}}$$

$$\underbrace{\text{ABB, WBB, WBBDOT)}}_{\text{OUTPUT}}$$

CALL GYROS

$$\overbrace{\text{(T, IENDF, WBB, WBBDOT, ABB,}}^{\text{INPUT}}$$

$$\underbrace{\text{DTHETA)}}_{\text{OUTPUT}}$$

CALL ACCEL

$$\overbrace{\text{(T, IENDF, ABB, WBB, WBBDOT,}}^{\text{INPUT}}$$

$$\underbrace{\text{DV)}}_{\text{INPUT}}$$

CALL ALTI

$$(\overbrace{\text{T, IENDF, ALT, VEL,}}^{\text{INPUT}}$$

$$\underbrace{\text{ALTO}}_{\text{OUTPUT}})$$

CALL RDR

$$(\text{T, }\overbrace{\text{IENDF, DTHETA, DV,}}^{\text{INPUT}}$$

$$\underbrace{\text{DTHETO, VDO}}_{\text{OUTPUT}})$$

CALL ACOMP

$$\overbrace{\text{(T, IENDF, DVO, DTHETO,}}^{\text{INPUT}}$$

$$\underbrace{\text{DVA}}_{\text{OUTPUT}})$$

CALL GCOMP

$$\overbrace{\text{(T, IENDF, DTHETO, DVA,}}^{\text{INPUT}}$$

$$\underbrace{\text{DTHETZ}}_{\text{OUTPUT}})$$

CALL ALG

$$\overbrace{\text{(T, IENDF, DTHETZ, DVA,}}^{\text{INPUT}}$$

$$\underbrace{\text{DVN, DCM}}_{\text{OUTPUT}}$$

CALL LLN

$$\overbrace{\text{(T, IENDF, DVN, ALTO, DCM,}}^{\text{INPUT}}$$

$$\underbrace{\text{NAVLAT, NAVLON, NAVV, NAVH, NAVP, NAVR, NAVHD)}}_{\text{OUTPUT}}$$

CALL EVL

$$\overbrace{\text{(T, IENDF, LAT, LON, ALT, VEL, DVT, PITCH,}}^{\text{INPUT (ALL)}}$$

ROLL, YAW, WONDER, NAVLAT, NAVLON,

NAVV, NAVH, NAVP, NAVR, NAVHD)

After each of the CALL statements, a dummy variable IF1, the switch IENDF returned from each module, is tested. If IF1 = 1, a flag, SIMEND = 1, is set, otherwise the next CALL statement is executed. (This is true for all subroutines except the trajectory module.)

The four modules, TRAJ, ENV, GYRO, ACCEL, are called at a fast cycle -- DT; while the remaining seven modules are called at the slow cycle -- DTSLOW.

The following is an alphabetical list giving definitions to the CALL statement variables.

Variable	Units	Data Type	Description
$\overline{\text{AB}}$	ft/s^2	REAL	specific force vector in body frame (from TRJ module)
$\overline{\text{ABB}}$	ft/s^2	REAL	specific force vector in body frame (includes vibrations from ENV module)
ALT	ft	REAL	altitude above sea level (from TRJ module)
ALTO	ft	REAL	indicated altitude above sea level from the altimeter
$\overset{*}{\text{DCM}}$	unity	REAL	direction cosine matrix
$\overline{\text{DTHETA}}$	rad	REAL	vector components of the quantized integrated change in the X, Y, and Z gyro input angles from GYRO module

Variable	Units	Data Type	Description
DTHETO	rad	REAL	vector components of the quantized integrated change in X, Y, Z gyro input angles in body frame from RDR module
DTHETZ	rad	REAL	vector components of the compensated quantized integrated change in X, Y, Z gyro input angles in body frame from GCOMP
DV	ft/s	REAL	vector component of the quantized integral of specific force in the body frame (from ACCEL module)
DVA	ft/s	REAL	vector components of the compensated quantized integral of specific force in the body frame (from ACOMP)
DVN	ft/s	REAL	vector components of the quantized integral of specific force in the inertial frame (from ALG module)
DVO	ft/s	REAL	vector components of the quantized nominally in the body frame (from RDR module)
DVT	ft/s	REAL	vector components of "true" integral of specific force in ENU frame (from TRAJ module)
HDING	rad	REAL	heading (YAW minus WANDER angle) from TRAJ
IENDF	logical	INTEGER	last-pass indicator all modules
LAT	rad	REAL	geodetic latitude from TRAJ
LON	rad	REAL	geodetic longitude from TRAJ
MODPDT	s	REAL	module print interval from TRAJ
NAVH	ft	REAL	computed navigational altitude from LLN
NAVHD	deg	REAL	computed navigational heading from LLN
NAVLAT	deg	REAL	computed navigational latitude from LLN
NAVLON	deg	REAL	computed navigational longitude from LLN
NAVP	deg	REAL	computed navigational pitch from LLN

Variable	Units	Data Type	Description
NAVR	deg	REAL	computed navigational roll
$\overline{\text{NAVV}}$	ft/s	REAL	computed navigational velocity vector in ENU frame
PITCH	rad	REAL	pitch angle (second rotation)
PRNTSW	logical	REAL	print switch (0 - no printout, printout otherwise)
ROLL	rad	REAL	roll angle (first rotation)
T	s	REAL	current simulation time
$\overline{\text{VEL}}$	ft/s	REAL	velocity vector in body frame
$\overline{\text{WB}}$	rad/s	REAL	inertial-angular rate vector in body frame
$\overline{\text{WBB}}$	rad/s	REAL	inertial-angular rate vector in body frame (includes vibrations from ENV module)
$\overline{\text{WBBDOT}}$	rad/s^2	REAL	inertial-angular acceleration vector in body frame (includes vibrations from ENV module)
WANDER	rad	REAL	wander angle (clockwise from north about the local vertical up axis)
YAW	rad	REAL	yaw angle (third rotation)

The following logic is performed at the end of each sequencer operating cycle.

Check for simulation end, If IENDF = 1, STOP

If T > TEND or If SIMEND > 0, set IENDF = 1 and make one last pass through each of the modules.

Increment simulation time, and correct for roundoff error.

$$T = T + DT$$

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

If T = 0, write '***START SIMULATION***'

At end of simulation, write '***SIMULATION COMPLETE AT (T at end) SEC'

(7) Subroutines Called

No subroutines are called.

2.3.2 INTERIM TRAJECTORY MODULE (TRAJ)

(1) General Description

This module is a replacement module for the old trajectory module, the latter of which interfaces the AFAL PROFGEN (profile generator) program with the INS Simulator. It is used to generate test case trajectory data in the absence of the PROFGEN program. Essentially, the function of this module is twofold:

- The generation of properly coordinated dynamic data to drive the simulated gyro, accelerometer, and altimeter hardware modules.
- The provision of navigational reference information (viz., "true" position and velocity and attitude) to evaluate the performance of the entire system.

(2) Trajectory Module Flow Diagram

The general logic flow of the TRJ module is shown in Figure 1.

An expanded diagram of the Print Module and print control is included in Section 2.2 of Volume III.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 20

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.01	s	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	logical	not used

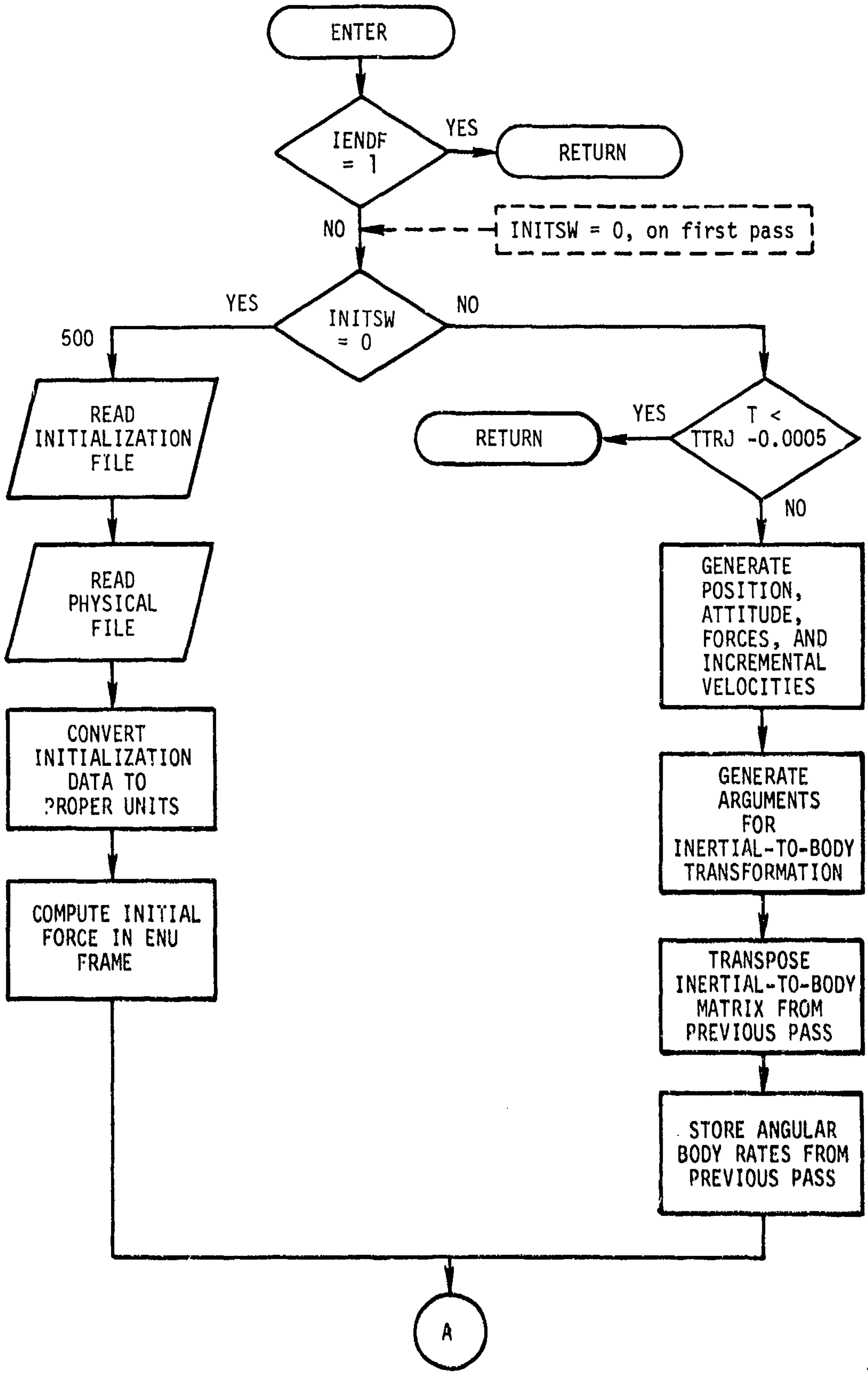

Figure 1. Trajectory module (sheet 1 of 2).

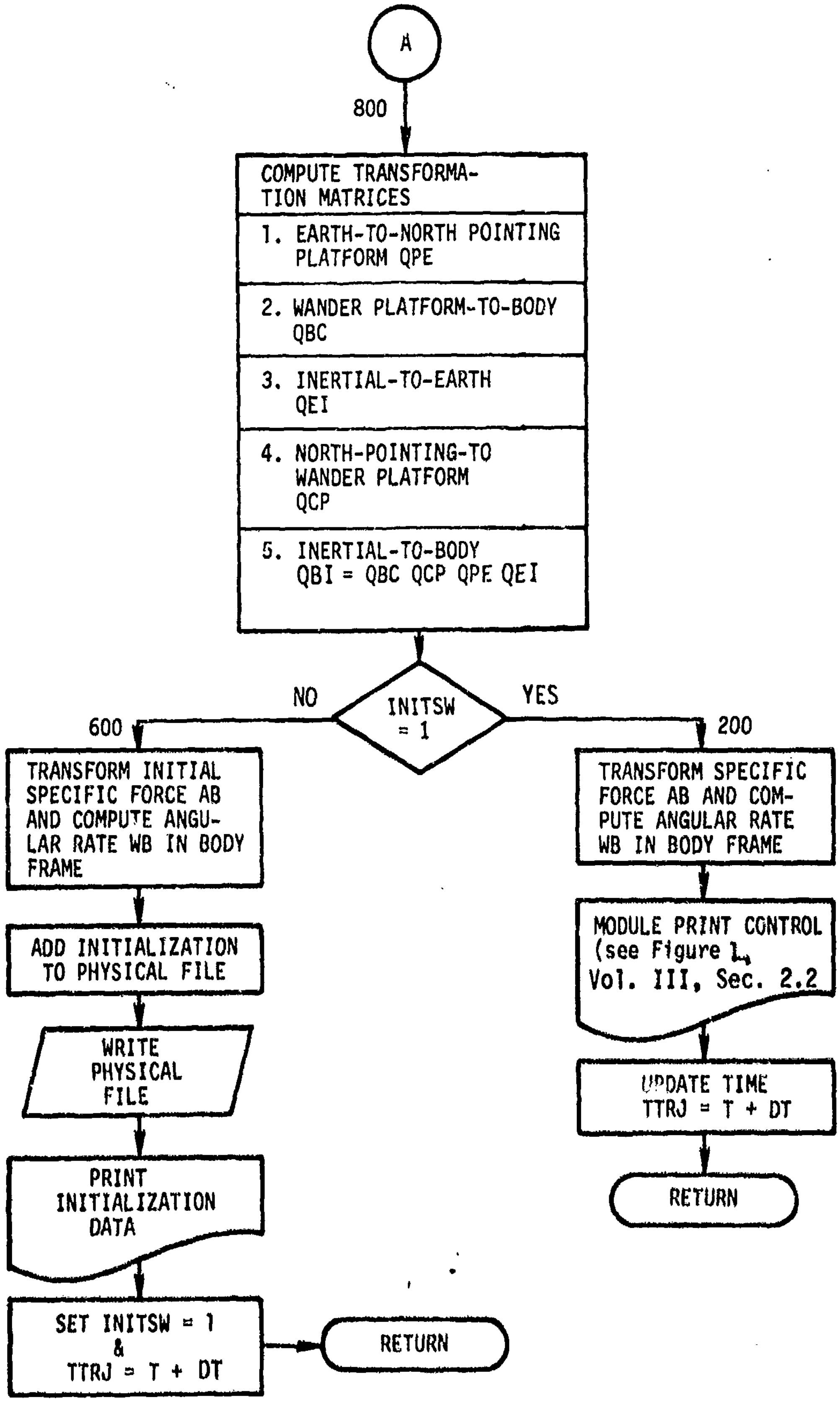

Figure 1. Trajectory module (sheet 2 of 2).

Index	Variable	Default Value	Units	Description
4	XFILE	6	logical	FORTRAN unit number for printout
5	ILAT	0.0	deg	initial latitude
6	ILONG	0.0	deg	initial longitude
7	IALT	0.0	ft	initial altitude above sea level
8	IVEL(1)	0.0	ft/s	initial velocity (VX)-EAST
9	IVEL(2)	0.0	ft/s	initial velocity (VY)-NORTH
10	IVEL(3)	0.0	ft/s	initial velocity (VZ)-UP
11	IPITCH	0.0	deg	initial pitch angle
12	IYAW	0.0	deg	initial yaw angle
13	IROLL	0.0	deg	initial roll angle
14	MODPDT	6.0	s	module print interval
15	IWANDR	0.0	deg	initial wander angle

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-04	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	LAT	0.0	deg	initial latitude

Index	Variable	Default Value	Units	Description
6	LON	0.0	deg	initial longitude
7	WANDER	0.0	deg	initial wander angle
8	ALT	0.0	ft	initial altitude above sea level
9	ROLL	0.0	deg	initial roll angle
10	PITCH	0.0	deg	initial pitch angle
11	YAW	0.0	deg	initial yaw angle
12	RDOT	0.0	rad/s	first time derivative of roll
13	PDOT	0.0	rad/s	first time derivative of pitch
14	YDOT	0.0	rad/s	first time derivative of yaw
15	VEL(1)	0.0	ft/s	initial east velocity wrt the earth (VX)
16	VEL(2)	0.0	ft/s	initial north velocity wrt the earth (VY)
17	VEL(3)	0.0	ft/s	initial up velocity wrt the earth (VZ)
18	AB(1)	0.0	ft/s^2	longitudinal specific force in body frame (AX)
19	AB(2)	0.0	ft/s^2	lateral specific force in body frame (AY)
20	AB(3)	0.0	ft/s^2	normal specific force in body frame (AZ)

2.3.3 ENVIRONMENT MODULE (ENV)

(1) General Description

The function of the ENV module is to simulate the random vehicle dynamics (e.g., translational acceleration and angular rates) from sources such as air turbulence. The algorithm used in this module takes a statistical representation, in the form of power-spectral densities (PSDs), of linear and angular displacement vibrations, then computes the appropriate random displacements (linear and angular) as functions of time. The algorithm's numerical differentiation of the suitable random displacement function—twice for translational acceleration, once for angular rate—effectively simulates these random vibrations. As outputs of the ENV module, the translational and angular vibrations are subsequently used, in combination with the dynamics transferred by the trajectory module, as the forcing functions of the INSS system.

Although the nominal (default) initialization file for this module uses the PSD characteristics of a typical low-altitude B-1 (1-ft/s gust standard deviation) mission (sensed in the forward avionics bay randome), the user may establish alternative PSD envelope characteristics in this file. An additional feature of this module is a vibration on/off switch (VIBSW), whose value is specified in the initialization file (defaulted to the "off" position). When vibration considerations are necessary, simulations require approximately twice as much computer time as would be the case if the ENV module were bypassed.

(2) Environment Module Flow Diagram

The general logic structure of the ENV module is displayed in Figure 1. Internal checks indicating either that the simulation termination time has been reached or that insufficient time has elapsed for the next iteration cycle will force an immediate return to the main (sequencer) program. If the vibration switch is set to the off position (VIBSW = 0), the only calculation performed before returning control

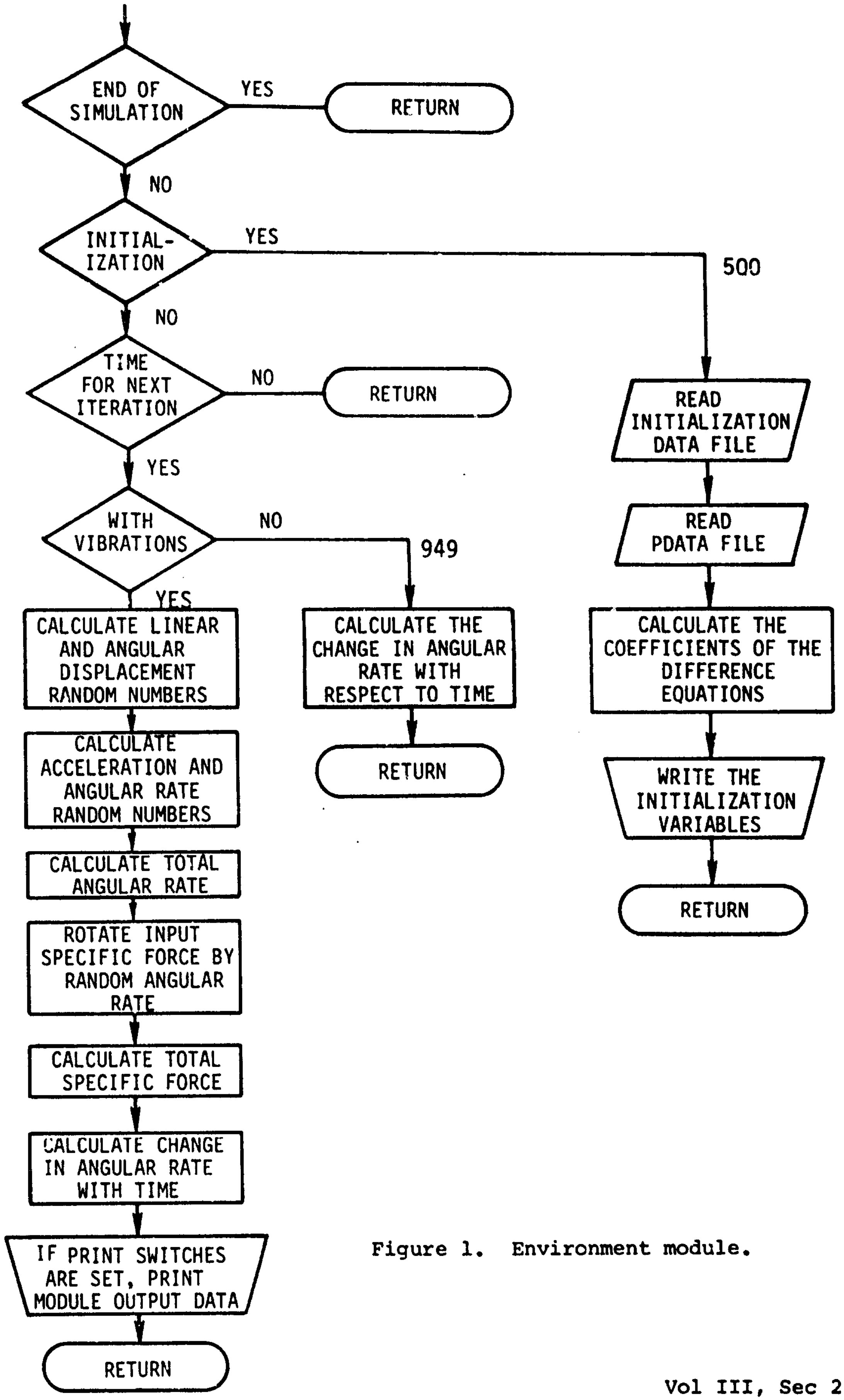

Figure 1. Environment module.

to the sequencer is the determination of the inertial angular-acceleration components in the body frame. The first initialization pass through this module (INITSW = 0) causes the initialization (IFILE) and physical (PDATA) data files to be read, coefficients of the difference equations to then be calculated, and initialization variables to be written before finally returning to the sequencer's control. The calculations invoked by subsequent passes through this module are shown in the accompanying flow chart (see Figure 1).

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 30

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.01	s	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	logical	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	S(1)	4.0	unity	number of peaks for each PSD (5 peaks per PSD maximum allowed): normal — linear displacement vibration
6	S(2)	3.0	unity	number of peaks for each PSD (5 peaks per PSD maximum allowed): lateral — linear displacement vibration
7	S(3)	0.0	unity	number of peaks for each PSD (5 peaks per PSD maximum allowed): longitudinal — linear displacement vibration
8	S(4)	3.0	unity	number of peaks for each PSD (5 peaks per PSD maximum allowed): pitch — angular displacement vibration
9	S(5)	4.0	unity	number of peaks for each PSD (5 peaks per PSD maximum allowed): yaw — angular displacement vibration
10	S(6)	1.0	unity	number of peaks for each PSD (5 peaks per PSD maximum allowed): roll — angular displacement vibration

Variables 11 - 40 are peak amplitudes of the six vibration PSD plots, with up to five peaks per plot allowed in the program. These amplitudes are for PSD plots normalized to unit wind gust intensity.

Index	Variable	Default Value	Units	Description
11	PARMAP(1)	136031.E-9	ft^2/Hz	normal-linear displacement vibration
12	PARMAP(2)	15488.E-9	ft^2/Hz	
13	PARMAP(3)	1615.E-9	ft^2/Hz	
14	PARMAP(4)	17.5E-9	ft^2/Hz	
15	PARMAP(5)	0.0	ft^2/Hz	
16	PARMAP(6)	10304.E-9	ft^2/Hz	lateral-linear displacement vibration
17	PARMAP(7)	59.E-9	ft^2/Hz	
18	PARMAP(8)	3.08E-9	ft^2/Hz	
19	PARMAP(9)	0.0	ft^2/Hz	
20	PARMAP(10)	0.0	ft^2/Hz	
21	PARMAP(11)	0.0	ft^2/Hz	longitudinal-linear displacement vibration
22	PARMAP(12)	0.0	ft^2/Hz	
23	PARMAP(13)	0.0	ft^2/Hz	
24	PARMAP(14)	0.0	ft^2/Hz	
25	PARMAP(15)	0.0	ft^2/Hz	
26	PARMAP(16)	262000.E-12	rad^2/Hz	pitch-angular displacement vibration
27	PARMAP(17)	64088.5E-12	rad^2/Hz	
28	PARMAP(18)	13226.E-12	rad^2/Hz	
29	PARMAP(19)	0.0	rad^2/Hz	
30	PARMAP(20)	0.0	rad^2/Hz	

Index	Variable	Default Value	Units	Description
31	PARMAP(21)	241902.5E-12	rad^2/Hz	yaw-angular displacement vibration
32	PARMAP(22)	24001.5E-12	rad^2/Hz	
33	PARMAP(23)	288.5E-12	rad^2/Hz	
34	PARMAP(24)	41.E-12	rad^2/Hz	
35	PARMAP(25)	0.0	rad^2/Hz	
36	PARMAP(26)	72.E-6	rad^2/Hz	roll-angular displacement vibration
37	PARMAP(27)	0.0	rad^2/Hz	
38	PARMAP(28)	0.0	rad^2/Hz	
39	PARMAP(29)	0.0	rad^2/Hz	
40	PARMAP(30)	0.0	rad^2/Hz	

Variables 41 - 70 are half-bandwidths of PSD peaks.

Index	Variable	Default Value	Units	Description
41	PARMWH(1)	3.0	rad/s	normal-linear displacement vibration
42	PARMWH(2)	3.0	rad/s	
43	PARMWH(3)	6.0	rad/s	
44	PARMWH(4)	4.0	rad/s	
45	PARMWH(5)	0.0	rad/s	
46	PARMWH(6)	3.5	rad/s	lateral-linear displacement vibration
47	PARMWH(7)	8.0	rad/s	
48	PARMWH(8)	6.0	rad/s	
49	PARMWH(9)	0.0	rad/s	
50	PARMWH(10)	0.0	rad/s	

Index	Variable	Default Value	Units	Description
51	PARMWH(11)	0.0	rad/s	longitudinal-linear displacement vibration
52	PARMWH(12)	0.0	rad/s	
53	PARMWH(13)	0.0	rad/s	
54	PARMWH(14)	0.0	rad/s	
55	PARMWH(15)	0.0	rad/s	
56	PARMWH(16)	3.3	rad/s	pitch-angular displacement vibration
57	PARMWH(17)	2.0	rad/s	
58	PARMWH(18)	6.0	rad/s	
59	PARMWH(19)	0.0	rad/s	
60	PARMWH(20)	0.0	rad/s	
61	PARMWH(21)	3.0	rad/s	yaw-angular displacement vibration
62	PARMWH(22)	5.0	rad/s	
63	PARMWH(23)	9.0	rad/s	
64	PARMWH(24)	3.0	rad/s	
65	PARMWH(25)	0.0	rad/s	
66	PARMWH(26)	2.0	rad/s	roll-angular displacement vibration
67	PARMWH(27)	0.0	rad/s	
68	PARMWH(28)	0.0	rad/s	
69	PARMWH(29)	0.0	rad/s	
70	PARMWH(30)	0.0	rad/s	

Variable 71 - 100 are center frequencies of PSD peaks.

Index	Variable	Default Value	Units	Description
71	PARMO(1)	13.5	rad/s	normal-linear displacement vibration
72	PARMO(2)	21.0	rad/s	
73	PARMO(3)	32.3	rad/s	
74	PARMO(4)	61.5	rad/s	
75	PARMO(5)	0.0	rad/s	
76	PARMO(6)	16.0	rad/s	lateral-linear displacement vibration
77	PARMO(7)	34.0	rad/s	
78	PARMO(8)	68.0	rad/s	
79	PARMO(9)	0.0	rad/s	
80	PARMO(10)	0.0	rad/s	
81	PARMO(11)	0.0	rad/s	longitudinal-linear displacement vibration
82	PARMO(12)	0.0	rad/s	
83	PARMO(13)	0.0	rad/s	
84	PARMO(14)	0.0	rad/s	
85	PARMO(15)	0.0	rad/s	
86	PARMO(16)	13.5	rad/s	pitch-angular displacement vibration
87	PARMO(17)	21.0	rad/s	
88	PARMO(18)	32.3	rad/s	
89	PARMO(19)	0.0	rad/s	
90	PARMO(20)	0.0	rad/s	

Index	Variable	Default Value	Units	Description
91	PARMO(21)	3.0	rad/s	yaw-angular displacement vibration
92	PARMO(22)	15.0	rad/s	
93	PARMO(23)	33.0	rad/s	
94	PARMO(24)	67.8	rad/s	
95	PARMO(25)	0.0	rad/s	
96	PARMO(26)	2.0	rad/s	roll-angular displacement vibration
97	PARMO(27)	0.0	rad/s	
98	PARMO(28)	0.0	rad/s	
99	PARMO(29)	0.0	rad/s	
100	PARMO(30)	0.0	rad/s	
101	MODPDT	6.0	logical	module print interval
102	VIBSW	0.0	logical	vibration switch (0 - module bypassed)
103	GUSTLA	1.0	ft^2/s^2	lateral — wind-gust intensity variable
104	GUSTLO	0.0	ft^2/s^2	longitudinal — wind-gust intensity variable
105	GUSTNR	1.0	ft^2/s^2	normal — wind-gust intensity variable

(b) <u>Common Initialization File (PFILE)</u>

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	------	not used

(4) Call-Line Data

$$\overbrace{(T,\ IENDF,\ \overline{AB},\ \overline{WB},}^{\text{INPUT}}\ \underbrace{\overline{ABB},\ \overline{WBB},\ \overline{WBBDOT})}_{\text{OUTPUT}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
$\overline{AB}$	ft/s^2	REAL	specific-force vector in body frame
$\overline{WB}$	rad/s	REAL	angular rate vector of body wrt inertial space in body frame

(b) Call-Line Output

Variable	Units	Data Type	Description
$\overline{ABB}$	ft/s^2	REAL	specific-force vector in body frame with vibration
$\overline{WBB}$	rad/s	REAL	angular rate vector of body wrt inertial space in body frame with vibration included
$\overline{WBBDOT}$	rad/s^2	REAL	angular acceleration vector of body wrt inertial space in body frame with vibration included

(5) Formulation

(a) Initialization Function

The switches INITSW, IN1, and IN2 are initialized to zero in DATA statements. The following functions are performed on the first pass, if INITSW = 0 and IENDF = 0, and the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE.

In order to calculate the coefficients of the difference equation form of the solution of the normal state space equation (Vol. II, Sec. 4 for a theoretical discussion), the following (up to 30) terms are computed for no more than five allowable PDS peaks for each of three linear (normal, lateral, or longitudinal) and three angular (pitch, yaw, or roll) displacement vibration types.

The number of PSD peaks are specified by S(J), where subscript J identifies the displacement vibration type.

The first term in each of two uncorrelated, zero mean, unity variance Gaussian random sequences, ETA and ZETA, are computed for each of the possible 30 PSD peaks.

Then, the following variables for each of the K-PSD peaks (S PSD's times 6 displacement types) are defined. First, multiply the PSD peak amplitude per unit wind-gust intensity by gust intensity variance product corrected for units.

$$AP = 2\pi * PARMAP(K) * GUST(J)$$

where

J = 1 to 6 for the six displacement vibration types

with

GUST(1) = GUSTNR, the normal-linear vibration.

GUST(2) = GUSTLA, the lateral-linear vibration.

GUST(3) = GUSTLO, the longitudinal-linear vibration.

GUST(4) = GUSTNR, the effect on pitch-angular vibration of the normal-linear vibration.

GUST(5) = GUSTLA, the effect on yaw-angular vibration of the lateral-linear vibration.

GUST(6) = GUSTLA, the effect on roll-angular vibration of the lateral-linear vibration.

Then, define the half-power bandwidth to central frequency ratio of each PSD peak

$$A = WH/WO$$

where

WH = PARMWH(K)

WO = PARMWO(K)

Now, the undamped natural frequency is

$$WN = WO*\left(2 - (1 - A^2/2)^2\right)^{1/4}$$

The damping ratio is

$$E = \left((1 - (WO/WN)^2/2\right)^{1/2}$$

and the filter gain is

$$F = \left(1 - (1 - 2E^2)^2\right)*AP$$

Initialize the displacements of previous pass for the K-PSD peaks as

X1PRV(K) = 0.0

X2PRV(K) = 0.0

Calculate the resonant frequency

$$WR = WN*(1 - E^2)^{1/2}$$

For convenience and economy of notation, the following variables are defined

$$Z = E*WN$$

$$E1 = EXP(-2*Z*DT)$$

$$C = ASIN(E)$$

$$\begin{aligned} U = {} & (1 - E1)/(4*Z*WR^2) \\ & + (E1*(Z*COS(2*WR*DT) - WR*SIN(2*WR*DT)))/(4*WN^2*WR^2) \\ & - E/(4*WN*WR^2) \end{aligned}$$

$$\begin{aligned} R = {} & (1 - E1)/((4*E*WR^2)/WN) \\ & - (E1*(Z*COS(2*WR*DT + 2*C) - WR*SIN(2*WR*DT + 2*C)))/4*WR^2) \\ & + (Z*COS(2*C) - WR*SIN(2*C))/(4*WR^2) \end{aligned}$$

$$\begin{aligned} V = {} & (E1 - 1)*SIN(C)/(4*E*WR^2) \\ & - (E1*(Z*SIN(2*WR*DT + C) + WR*COS(2*WR*DT + C)))/4*WN*WR^2) \\ & + (WR*COS(C) + Z*SIN(C))/(4*WN*WR^2) \end{aligned}$$

Finally, the required coefficients of the difference equation form of the solution of the normal state space equation are computed. For each PSD peak the set of coefficients computed is

$$B = (|U - (V^2/R)|)^{1/2}*F^{1/2}*WN^2$$

$$H = (V/R^{1/2})*F^{1/2}*WN^2$$

$$P = R^{1/2}*F^{1/2}*WN^2$$

$$P12 = (SIN(WR*DT)*EXP(-Z*DT))/WR$$

$$P22 = COS(WR*DT) - (Z/WR)*SIN(WR*DT)*EXP(-Z*DT)$$

$$P11 = 2*Z*P12 + P22$$

$$P21 = -WN^2*P12$$

where B, H, P are the coefficients of the noise terms ETA and ZETA and P11, P12, P21, and P22 are elements of the transition matrix.

The K'th set of coefficients is stored as the K'th elements in a set of coefficient array variables.

COEF1(K) = B

COEF2(K) = H

COEF3(K) = P

COEF4(K) = P12

COEF5(K) = P22

COEF6(K) = P11

COEF7(K) = P21

Now set the specific force, the body inertial-angular rate, and the body inertial-angular acceleration in the body frame as

$$\overline{WBB} = \overline{WB}$$

$$\overline{ABB} = \overline{AB}$$

$$\overline{WBBDOT} = (0.0, 0.0, 0.0)$$

PRINT out the initialization data.

Set INITSW = 1 and increment the simulation time

TENV = T + DT

and return to the main program.

(b) General Function

If the vibration switch

VIBSW < 0

then the vibration computation is bypassed and the inertial angular rate of the previous pass is defined as

$$\overline{WBBOLD} = \overline{WBB}$$

and then

$$\overline{WBB} = \overline{WB}$$

$$\overline{ABB} = \overline{AB}$$

$$\overline{WBBDOT} = (\overline{WBB} - \overline{WBBOLD})/DT$$

The results are printed, the simulation time is incremented

$$TENV + DT$$

And there is a return to the main program.

If the vibration switch

$$VIBSW \geq 1$$

The module generates sequences of random numbers ETA and ZETA, to force the linear difference equations, and Y is initialized to zero for each of the PSDs.

The random processes X1(I) and X2(I) are generated for each of the I spectral density peaks for each single displacement vibration type

X1(I) = P11*X1PRV(I) + P12*X2PRV(I) + H*ETA + B*ZETA

X2(I) = P21*X1PRV(I) + P22*X2PRV(I) + P*ETA and solved for the random process, X1(I). On the first pass after initialization, when IN2 = 0, each value of X1(I) is divided by the module operating frequency to obtain a velocity, and that velocity is summed over the power spectral density peaks in a summation on the index variable I

$$Y = Y + X1(I)/DT$$

On succeeding passes, the above summation is replaced with

$$Y = Y + (X1(I) - X1PRV(*))/DT$$

The Y variable is controlled in a summation with index J which identifies the J'th value of Y as the J'th vibration type. The J'th value of Y is inserted as the J'th element of the array variable RAND via

$$RAND(J) = Y$$

for each of the six displacement vibration types.

The values X1(I) and X2(I) are stored in array variables representing the previous values for the next solution loop

$$X1PRV(K) = X1(I)$$

$$X2PRV(K) = X2(I)$$

and IN2 is set to 1. This completes the vibration generation.

Now the environment output, angular rates, and specific forces, which include vibration-induced motions, are calculated. First, set the body angular rate with vibrations from previous pass to

$$\overline{WBBOLD} = \overline{WBB}$$

Then, the three linear-displacement vibrations

$$\overline{\mathrm{DELVB}} = \text{first three components of RAND}$$

Now, compute the change in specific force due the vibration

$$\overline{\mathrm{DELAB}} = (\overline{\mathrm{DELVB}} - \overline{\mathrm{DELVBP}})/\mathrm{DT}$$

where $\overline{\mathrm{DELVBP}}$ is the $\overline{\mathrm{DELVB}}$ of the previous pass.

Now redefine

$$\overline{\mathrm{DELVBP}} = \overline{\mathrm{DELVB}}$$

the $\overline{\mathrm{DELVBP}}$ as the $\overline{\mathrm{BELVB}}$ of the previous pass for the next pass.

The three angular-displacement vibrations are defined as

$$\overline{\mathrm{DELWB}} = \text{last three components of RAND}$$

i.e., the angular-displacement vibrations of pitch, yaw, and roll, respectively.

The components of the angular-displacement (pitch, yaw, and roll) vibrations are added to the angular rates in the body frame

$$\mathrm{WBB}(1) = \mathrm{WB}(1) + \mathrm{DELWB}(3)$$

$$\mathrm{WBB}(2) = \mathrm{WB}(2) + \mathrm{DELWB}(1)$$

$$\mathrm{WBB}(3) = \mathrm{WB}(3) + \mathrm{DELWB}(2)$$

The transformation matrix that establishes the turbulence-induced change in body attitude is computed from

$$\overset{*}{\text{QBBCHG}} = \left[\begin{array}{c|c|c} 1.0 & \begin{array}{c}\text{(DELWB(2)}\\ \text{+ DELWBP(2))/2DT}\end{array} & \begin{array}{c}\text{(DELWB(1)}\\ \text{+ DELWBP(1))/2DT}\end{array} \\ & & \\ \begin{array}{c}\text{- (DELWB(2)}\\ \text{+ DEWBP(2))/2DT}\end{array} & 1.0 & \begin{array}{c}\text{(DELWB(3)}\\ \text{+ DELWBP(3))/2DT}\end{array} \\ & & \\ \begin{array}{c}\text{(DELWB(1)}\\ \text{+ DELWBP(1))/2DT}\end{array} & \begin{array}{c}\text{-(DELWB(3)}\\ \text{+ DELWBP(3))/2DT}\end{array} & 1.0 \end{array}\right]$$

where the terms of the matrix are the vibration-induced rate, and its previous pass value averaged and used over the time increment.

Then, the total change in body attitude is given by

$$\overset{*}{\text{QBB}} = \overset{*}{\text{QBBCHG}} * \overset{*}{\text{QBBPRV}}$$

where $\overset{*}{\text{QBBPRV}} = \overset{*}{\text{QBB}}$ of the previous pass. On the first pass after initialization

$$\overset{*}{\text{QBBPRV}} = \overset{*}{\text{I}}$$

from a computation time initialization.

The terms of the vibration-induced angular rate, $\overline{\text{DELWB}}$, from the current pass are stored as $\overline{\text{DELWBP}}$

$$\overline{\text{DELWBP}} = \overline{\text{DELWB}}$$

Now, the attitude matrix, $\overset{*}{QBB}$, is orthonormalized. First, compute the cross-product matrix, $\overset{*}{QTQ}$,

$$\overset{*}{QTQ} = ({}^{3}\overset{*}{I} - \overset{*}{QBB}^{T} * \overset{*}{QBB})/2$$

where $\overset{*}{I}$ is the identify matrix.

Orthonormalization of $\overset{*}{QBB}$ is complete with the computation

$$\overset{*}{QBB} = \overset{*}{QBB} * \overset{*}{QTQ}$$

Next, the rotated specific force is computed

$$\overline{ABB} = \overset{*}{QBB} * \overline{AB}$$

and the nominal gravity times the linear-displacement vibrations for longitudinal, lateral, and normal displacements are added to the body frame components of the specific force without vibration

$$ABB(1) = ABB(1) + DELAB(3)*G$$

$$ABB(2) = ABB(2) + DELAB(2)*G$$

$$ABB(3) = ABB(3) + DELAB(1)*G$$

The attitude matrix, $\overset{*}{QBB}$, of the current pass is stored as $\overset{*}{QBBPRV}$ for use in the next pass.

$$\overset{*}{QBBPRV} = \overset{*}{QBB}$$

Finally, the angular accelerations, $\overline{WBBDOT}$, are calculated

$$\overline{WBBDOT} = (\overline{WBB} - \overline{WBBOLD})/T$$

Output is printed, and the simulation time is incremented

$$TENV = T + DT$$

and there is a return to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass, the title, "ENVIRONMENT INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for print control logic. The printed output is, as follows:

Variables	Units	Description
$\overline{AB}$	ft/s^2	specific-force vector in body frame without vibrations
$\overline{WB}$	rad/s	angular-rate vector in body frame without vibrations
$\overline{ABB}$	rad/s	specific-force vector in body frame with vibration included
$\overline{WBB}$	rad/s	angular-rate vector in body frame with vibration included
$\overline{WBB}$DOT	rad/s^2	angular-acceleration vector in body frame with vibration included

(7) Subroutines Called (See Section 2.3.15)

MXV = matrix by vector product

MXM = matrix multiplication

MTXM = matrix transpose by matrix product

GAUSS(AM,SD) = Gaussian random number generator function with mean, AM, and standard deviation, SD.

2.3.4 GYRO MODULE (GYROS)

(1) General Description

The GYRO module simulates the performance of three conventional single-degree-of-freedom (SDF) rate integrating gyros. Instrument kinematics are modeled on the basis of one of three available (user-specified) differential equations: a "performance" model (without gyro inertia or damping terms); a first-order differential equation (with only gyro damping); and a second-order differential equation (with both gyro damping and inertia). Also simulated in this module are the following gyro sensitivities:

- Acceleration sensitivity (both g and g-squared terms).
- Instrument misalignments.
- Anisoinertia.
- Output-axis coupling.
- Quantization levels.
- Scale-factor errors (both positive and negative).
- Scale-factor-error rate sensitivity (both positive and negative).
- Bias.
- Bias transient at turn-on.
- Random bias (exponentially correlated).

Although the user may select any alternate gyro parameters by modifying the attendant initialization data file, the nominal (default) parameters used by this module correspond to the CSDL 18 IRIG Mod B gyro in an analog rebalance loop.

(2) Gyro Module Flow Diagram

The general flow logic of the gyro module is illustrated in Figure 1.

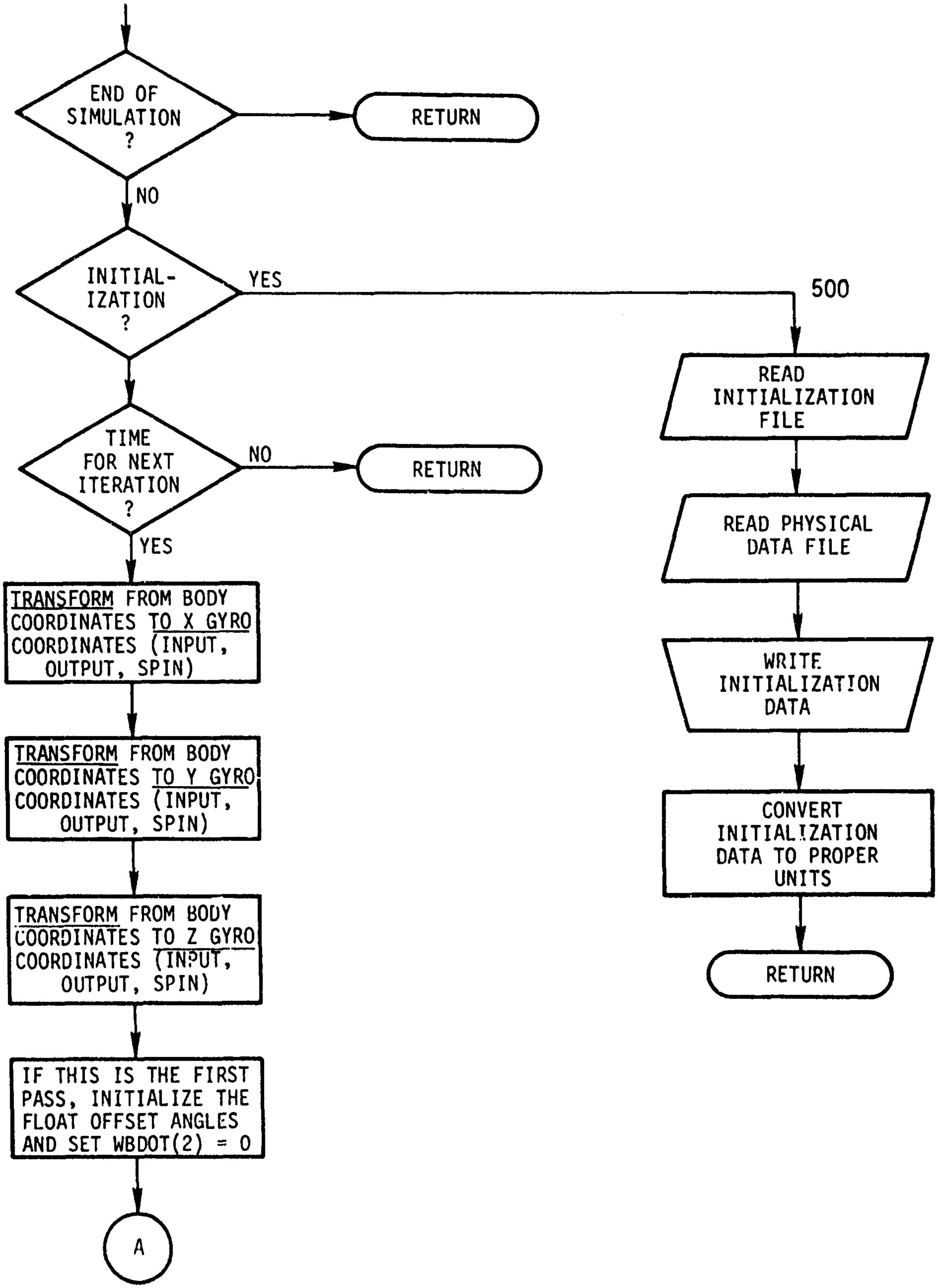

Figure 1. Gyro Module (sheet 1 of 2).

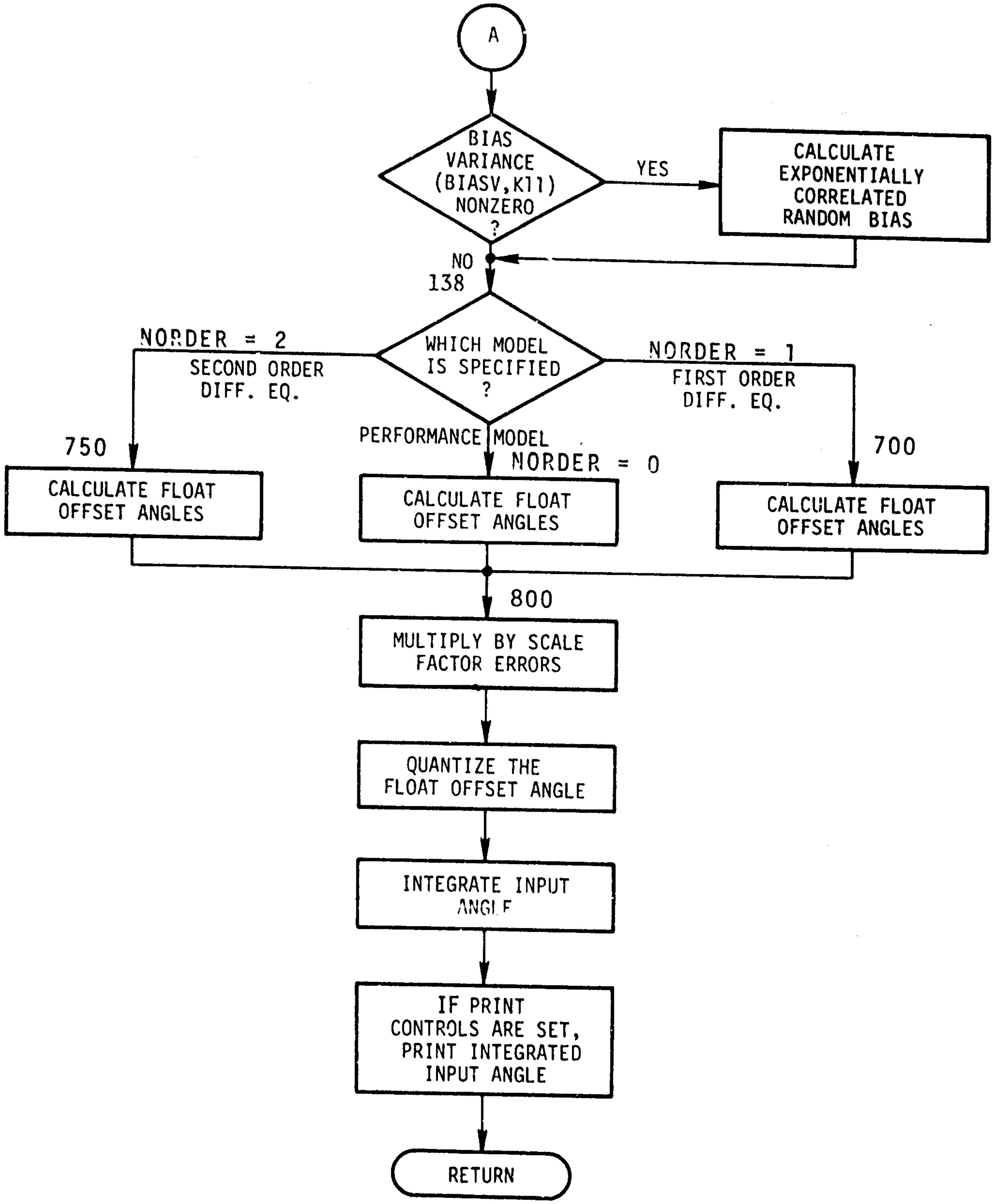

Figure 1. Gyro Module (sheet 2 of 2).

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 40

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.01	s	module operating timestep
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	-------	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	-------	not used
6	SPARE2	0.0	-------	not used
7	CO	7000000.0	dyn-cm/ rad/s	gyro damping coefficient
8-12	-------	---	-------	not used
13	QGBX(1)	1.0	unity	X-gyro transformation matrix from body coordinates to gyro (input, output, spin) coordinates
14	QGBX(2)	0.0	unity	
15	QGBX(3)	0.0	unity	
16	QGBX(4)	0.0	unity	
17	QGBX(5)	0.0	unity	
18	QGBX(6)	1.0	unity	
19	QGBX(7)	0.0	unity	
20	QGBX(8)	-1.0	unity	
21	QGBX(9)	0.0	unity	

Index	Variable	Default Value	Units	Description
22	QGBY(1)	0.0	unity	
23	QGBY(2)	1.0	unity	
24	QGBY(3)	0.0	unity	
25	QGBY(4)	1.0	unity	
26	QGBY(5)	0.0	unity	Y-gyro transformation matrix from body coordinates to gyro (input, output, spin) coordinates
27	QGBY(6)	0.0	unity	
28	QGBY(7)	0.0	unity	
29	QGBY(8)	0.0	unity	
30	QGBY(9)	-1.0	unity	
31	QGBZ(1)	0.0	unity	
32	QGBZ(2)	0.0	unity	
33	QGBZ(3)	1.0	unity	
34	QGBZ(4)	1.0	unity	
35	QGBZ(5)	0.0	unity	Z-gyro transformation matrix from body coordinates to gyro (input, output, spin) coordinates
36	QGBZ(6)	0.0	unity	
37	QGBZ(7)	0.0	unity	
38	QGBZ(8)	1.0	unity	
39	QGBZ(9)	0.0	unity	
40	H	151000.0	$gm\text{-}cm^2/s$	angular momentum of gyro wheels
41	--------	---	--------	not used
42	QUANT	0.00007	arcsec	quantization level of digitized change in angle

Index	Variable	Default Value	Units	Description
43	BIAS(1)	-0.31	deg/h	X-gyro bias levels
44	BIAS(2)	-0.31	deg/h	Y-gyro bias levels
45	BIAS(3)	-0.31	deg/h	Z-gyro bias levels
46	K	3100000000.0	$gm\text{-}cm^2/s^2$	rebalance loop elastic restraint
47	I	226.0	$gm\text{-}cm^2$	gyro output axis moment of inertia
48	DELI	14.0	$gm\text{-}cm^2$	gyro spin axis minus input axis moment of inertia
49	-------	---	-------	not used
50	KI(1)	1.12	deg/h/g	input axis acceleration-sensitive coefficients for X-gyro
51	KI(2)	1.12	deg/h/g	input axis acceleration-sensitive coefficients for Y-gyro
52	KI(3)	1.12	deg/h/g	input axis acceleration-sensitive coefficients for Z-gyro
53	KO(1)	-0.032	deg/h/g	output axis acceleration-sensitive coefficients for X-gyro
54	KO(2)	-0.032	deg/h/g	output axis acceleration-sensitive coefficients for Y-gyro
55	KO(3)	-0.032	deg/h/g	output axis acceleration-sensitive coefficients for Z-gyro
56	KS(1)	-0.62	deg/h/g	spin axis acceleration-sensitive coefficients for X-gyro
57	KS(2)	-0.62	deg/h/g	spin axis acceleration-sensitive coefficients for Y-gyro
58	KS(3)	-0.62	deg/h/g	spin axis acceleration-sensitive coefficients for Z-gyro
59	KII(1)	0.0	$deg/h/g^2$	input axis × input axis acceleration-squared sensitive coefficients for X-gyro
60	KII(2)	0.0	$deg/h/g^2$	input axis × input axis acceleration-squared sensitive coefficients for Y-gyro
61	KII(3)	0.0	$deg/h/g^2$	input axis × input axis acceleration-squared sensitive coefficients for Z-gyro
62	KSS(1)	-0.030	$deg/h/g^2$	spin axis × spin axis acceleration-squared sensitive coefficients for Y-gyro
63	KSS(2)	-0.030	$deg/h/g^2$	spin axis × spin axis acceleration-squared sensitive coefficients for Y-gyro
64	KSS(3)	-0.030	$deg/h/g^2$	spin axis × spin axis acceleration-squared sensitive coefficients for Z-gyro

Index	Variable	Default Value	Units	Description	
65	KIO(1)	0.0	$deg/h/g^2$	input axis × output axis acceleration-squared sensitive coefficient for	X-gyro
66	KIO(2)	0.0	$deg/h/g^2$		Y-gyro
67	KIO(3)	0.0	$deg/h/g^2$		Z-gyro
68	KIS(1)	0.21	$deg/h/g^2$	input axis × spin axis acceleration-squared sensitive coefficient for	X-gyro
69	KIS(2)	0.21	$deg/h/g^2$		Y-gyro
70	KIS(3)	0.21	$deg/h/g^2$		Z-gyro
71	KOS(1)	-0.0075	$deg/h/g^2$	output axis × spin axis acceleration-squared sensitive coefficient for	X-gyro
72	KOS(2)	-0.0075	$deg/h/g^2$		Y-gyro
73	KOS(3)	-0.0075	$deg/h/g^2$		Z-gyro
74	BIASV(1)	0.0	$(deg/h)^2$	variance of random bias (exponentially correlated) for	X-gyro
75	BIASV(2)	0.0	$(deg/h)^2$		Y-gyro
76	BIASV(3)	0.0	$(deg/h)^2$		Z-gyro
77	SFPO(1)	-20.0	ppm	positive scale-factor errors for	X-gyro
78	SFPO(2)	-20.0	ppm		Y-gyro
79	SFPO(3)	-20.0	ppm		Z-gyro
80	SFMO(1)	-54.0	ppm	negative scale-factor errors for	X-gyro
81	SFMO(2)	-54.0	ppm		Y-gyro
82	SFMO(3)	-54.0	ppm		Z-gyro

Index	Variable	Default Value	Units	Description
83	MODPDT	6.0	s	module print interval
84	ORDER	0.0	logical	order of differential equation model (0 = performance model, 1 = first-order differential equation, 2 = second-order differential equation)
85	TRANS1(1)	0.0	deg/h	exponential decay turn-on bias initial values for X-gyro
86	TRANS1(2)	0.0	deg/h	exponential decay turn-on bias initial values for Y-gyro
87	TRANS1(3)	0.0	deg/h	exponential decay turn-on bias initial values for Z-gyro
88	TRANTC(1)	200.0	s	exponents decay turn-on bias time constant for X-gyro
89	TRANTC(2)	200.0	s	exponents decay turn-on bias time constant for Y-gyro
90	TRANTC(3)	200.0	s	exponents decay turn-on bias time constant for Z-gyro
91	SFPI(1)	0.0	ppm/rad/s	positive scale-factor errors varying with rate for X-gyro
92	SFPI(2)	0.0	ppm/rad/s	positive scale-factor errors varying with rate for Y-gyro
93	SFPI(3)	0.0	ppm/rad/s	positive scale-factor errors varying with rate for Z-gyro
94	SFMI(1)	0.0	ppm/rad/s	negative scale-factor errors varying with rate for X-gyro
95	SFMI(2)	0.0	ppm/rad/s	negative scale-factor errors varying with rate for Y-gyro
96	SFMI(3)	0.0	ppm/rad/s	negative scale-factor errors varying with rate for Z-gyro
97	BIASTC	40.0	s	exponentially correlated random bias time constant (must be nonzero)

(b) <u>Common Initialization File (PFILE)</u>

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	-----	not used

(4) Call-Line Data

$$\overbrace{\text{(T, IENDF, WBB, WDOT, ABB,}}^{\text{INPUT}}$$

$$\underbrace{\text{DTHETA)}}_{\text{OUTPUT}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
$\overline{\text{WBB}}$	rad/s	REAL	angular rate vector of body wrt inertial space in body frame with vibration included
$\overline{\text{WDOT}}$	rad/s^2	REAL	angular acceleration vector of body wrt inertial space in body frame with vibration included (WBBDOT)
$\overline{\text{ABB}}$	ft/s^2	REAL	specific force in body frame with vibration included

(b) Call-Line Output

Variable	Units	Data Type	Description
DTHETA	rad	REAL	quantized integral of angular rate about X, Y, Z gyro input axes over the computation cycle, plus errors

(5) Formulation

(a) Initialization Function

The switch INITSW is initialized to zero in DATA statement. The following functions are performed on the first pass, if INITSW = 0 and IENDF = 0 and the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE
  and NORDER = ORDER.
- Convert initialization data to computational units

$$S\overline{P}O = S\overline{FP}O*1.E-6$$

$$S\overline{P}1 = S\overline{FP}1*1.E-6$$

$$S\overline{M}1 = S\overline{FM}1*1.E-6$$

$$S\overline{M}O = S\overline{FM}O*1.E-6$$

$$AQUANT = QUANT*\pi/6.48E5*H/DT/K$$

$$EX = EXP(-DT/BIASTC)$$

$$CO = CO*2.37E-6$$

$$DELI = DELI*2.37E-6$$

$$K = K*2.37E-6$$

$$I = I*2.37E-6$$

$$DTI = DT/I$$

$$H = H*2.37E-6$$

$$TR\overline{AN}S1 = TRANS1*4.86E-6$$

$$B\overline{IAS}A = (B\overline{IA}SV*(1 - EX^2))^{1/2}*4.85E-6$$

$$B\overline{IA}S = B\overline{IA}S*4.85E-6$$

In addition to converting to computational units, the following variables are tested and the switches K11 and K12 (which were initialized to zero in the DATA atatement) are reset.

If $\overline{BIASV} \neq 0$, set K11 = 1

If $\overline{KI} \neq 0$, set K12 = 1

If $\overline{KO} \neq 0$, set K12 = 1

If $\overline{KS} \neq 0$, set K12 - 1

If $\overline{KII} \neq 0$, set K12 = 1

If $\overline{KSS} \neq 0$, set K12 = 1

IF $\overline{KIO} \neq 0$, set K12 = 1

$\overline{KI}$ = KI*4.85E-6/G = 1

$\overline{KO}$ = KO*485E-6/G = 1

$\overline{KS}$ = KS*4.85E-6/G = 1

$\overline{KII}$ = KII*4.85E-6/G^2 = 1

$\overline{KSS}$ = KSS*485E-6/G^2 = 1

$\overline{KIO}$ = KIO*4.85E-6/G^2 = 1

Initialization is now complete, INITSW is set to 1, and the simulation time is incremented

$$TGYR = T + DT$$

Now, the routine returns to the main program.

(b) <u>General Functions</u>

The following functions are performed every module operating cycle.

The quantized integrated change in the input angle of the X, Y, Z-gyros, $\overline{THETA}$, is computed in turn.

First for the X-gyro, when I1 = 1, the coordinates of specific force, angular rate, and angular acceleration are transformed from (XYZ) body coordinates to (IOS) gyro coordinates

$$\overline{ABG} = \overset{*}{QGBX} * \overline{ABB}$$

$$WBG = \overset{*}{QGBX} * \overline{WBB}$$

$$\overline{WBDOT} = \overset{*}{QGBX} * \overline{WDOT}$$

Similarly for the Y-gyro, when I1 = 2,

$$\overline{ABG} = \overset{*}{QGBY} * \overline{ABB}$$

$$\overline{WBG} = \overset{*}{QGBY} * \overline{WBB}$$

$$\overline{WBDOT} = \overset{*}{QGBZ} * \overline{WDOT}$$

and for the Z-gyro, when I1 = 3

$$\overline{ABG} = \overset{*}{QGBZ} * \overline{ABB}$$

$$\overline{WBG} = \overset{*}{QGBZ} * \overline{WBB}$$

$$\overline{WBDOT} = \overset{*}{QGBZ} * \overline{WDOT}$$

Now, for tyros X, Y, and Z, when I1 = 1, 2, and 3 respectively, the following terms are calculated.

If on the first pass after initialization (K10 = 0), set

$$WBDOT(2) = 0.0$$

and the output axis value

$$THETA(I1) = WBG(1) * H/K$$

where WBG(1) is the angular rate along the input axis.

The bias levels are now calculated. Fisrt, if the bias variance is zero, BIASV(I1) = 0.0 (K11 = 0). Then, the random number generation is bypassed, and the random bias is set to the bias level

$$BIA = BIAS(I1)$$

Otherwise, a Gaussian random number with zero mean and a standard deviation equal to BI = BIAS(I1) is generated and the exponential correlated random bias, BIA, is calculated. The value of GBIAS for the particular gyro from the previous pass (on the first pass after the initialization, GBIAS = (0.0, 0.0, 0.0) is multiplied by the exponential function, EX, and added to the random number

GBIAS(I1) = GBIAS(I1)*EX + GAUSS(AM,BI)

where

EX = EXP(-DT/BIASTC)

Finally, the exponentially correlated random bias is

BIA = BIAS(I1) + GBIAS(I1)

The single-degree-of-freedom gyro module dynamic equations are solved for the float offset angles, THETA, for any one of three approximate forms:

- The performance model when NORDER = 0.
- The first-order model when NORDER = 1.
- The second-order model when NORDER = 2.

See Vol II, Sec 5.1.1 for complete description.

A quantity DEN is defined

$$DEN = K + DELI*(WBG(3)^2 - WBG(1)^2) + H*WBG(3)$$

where the subscripts 3 and 1 denote the spin and input axes, DEN consists of three torque terms. These terms are the rebalance loop elastic restraint torque, the anisoinertia coupling, and the cross-coupling torque per unit float offset.

Further we define M1 consisting of the torque terms: input, random bias, exponentially decaying turn-on bias, torque due to Coriolis effects, and output axis angular-acceleration torque

$$
\begin{aligned}
M1 \;&=\; H*(WBG(1) - BIA + TRANS1(I1)*EXP9\text{-}T/TRANTC(I1))) \\
&\quad + DELI*WBG(3)*WBG(1) - I*WBDOT(2)
\end{aligned}
$$

Additional torque terms in the numerator, denoted by M2 are computed for acceleration and acceleration-squared sensitive effects, if any of the coefficients are nonzero or when K12 = 1

$$
\begin{aligned}
M2 \;&=\; -H*(KI(I1)*ABG(1) + KS(I1)*ABG(3) + KO(I1)*ABG(2) \\
&\quad + KSS(I1)*ABG(3)^{2} + KII(I1)*ABG(1)^{2} \\
&\quad + KIS(I1)*ABG(1)*ABG(3) + KIO(I1)*ABG(1)*ABG(2) \\
&\quad + KOS(I1)*ABG(2)*ABG(3))
\end{aligned}
$$

Otherwise, M2 = 0.0.

If NORDER = 0, the performance model equation is solved for the float offset angle, THN

$$THN = (M1 + M2)/DEN$$

If NORDER = 1, from the first-order model equation, THN is

$$THN = THETA(I1) + DT*(-DEN*THETA(I1) + M1 + M2)/CO$$

If NORDER = 2, from the second-order model equation, THN is

$$THN = THETA(I1) + DT*THDOT(I1)$$

where THDOT(I1) is the previous pass value which is initially zero.

The term THDOT is calculated for use in the next pass from the expression

$$\mathrm{THDOT(I1) = THDOT(I1) + (M1 + M2 - CO*THDOT(I1) - DEN*THETA(I1))*DTI}$$

Finally, the quantized integral of angular rate about the gyro input axes, DTHETA, are calculated for the X, Y, or Z gyro depending upon the value of the index, I1.

First, the float offset angle is added to the residual output angle from previous quantization

$$\mathrm{TTHET(I1) = THN + TTHET(I1)}$$

Scale factor errors are accounted for in the case of positive offset angle by defining

$$\mathrm{T1 = 1 + SP0(I1) + SP1(I1)*THN*K/H}$$

or in the case of negative offset angle by defining

$$\mathrm{T1 = 1 + SM0(I1) + SM1(I1)*THN*K/H}$$

Using this definition to account for the scale-factor error, the integer variable,

$$\mathrm{ITHET = TTHET(I1)/(AQUANT*T1)}$$

is computed. It expresses how many quantum steps of angle, AQUANT, are contained ITHET and represents the increment in pulse count from a digital rebalance gyro.

When forming the integer pulse count, ITHET, any fractional part of a quantum step is dropped. This fractional part represents a residual float offset which should be retained for the next computation cycle. The residual offset angle is retrieved as

TTHET(I1) = TTHET(I1) - TT*T1*AQUANT

where

TT = FLOAT(ITHET) (Keeps the FORTRAN arithmetic in real variables).

The offset angle THN is renamed

THETA(I1) = THN

for use in the next THN computation.

Now, the desired output from the gyro module, DTHETA, is obtained for the X, Y, and Z gyros.

DTHETA(I1) = DTHETA(I1) + TT*AQUANT*DT*K/H

The DTHETA is printed and the THETA angle initialization switch is set to 1, KIO = 1. The simulation time is incremented

TGYR = T + DT

and the subroutine returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass the title "GYROSCOPE INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for print control logic. The printed output is, as follows

Variables	Units	Description
DTHETA,	rad	quantized integral of angular rate about the X, Y, Z gyro input axes over the computation cycle, plus errors

(7) <u>Subroutines Called</u> (See Section 2.3.15)

MXV = matrix by vector product.

GAUSS(AM,BI) = Gaussian random number generator with mean, AM, and standard deviation, BI.

2.3.5 LASER GYRO MODULE (GYROS)

(1) General Description

The laser gyro module simulates three body mounted laser gyros. The laser gyro module contains the following error sources:

- gyro input axis misalignment
- scale factor error
- scale factor turn-on transient
- gyro bias
- turn-on transient drift
- angular random walk
- white angular noise
- angular quantization

Although the user may select alternate gyro parameters by modifying the gyro module initialization file (IFILE), the nominal (default) values are representative of the Honeywell GG1300 laser gyro.

(2) Gyro Module Flow Diagram

Figure 1 is a flow diagram of the laser gyro module. The module is initialized on the first pass through the routine. The laser gyro module initialization data file (IFILE) is read at initialization and the input data is converted into internal program units. The program then performs the normal module operations. An analytical description of the laser gyro module equations is given in Volume II, Section 5.2.2.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 40

FORTRAN format: I5, F20.10

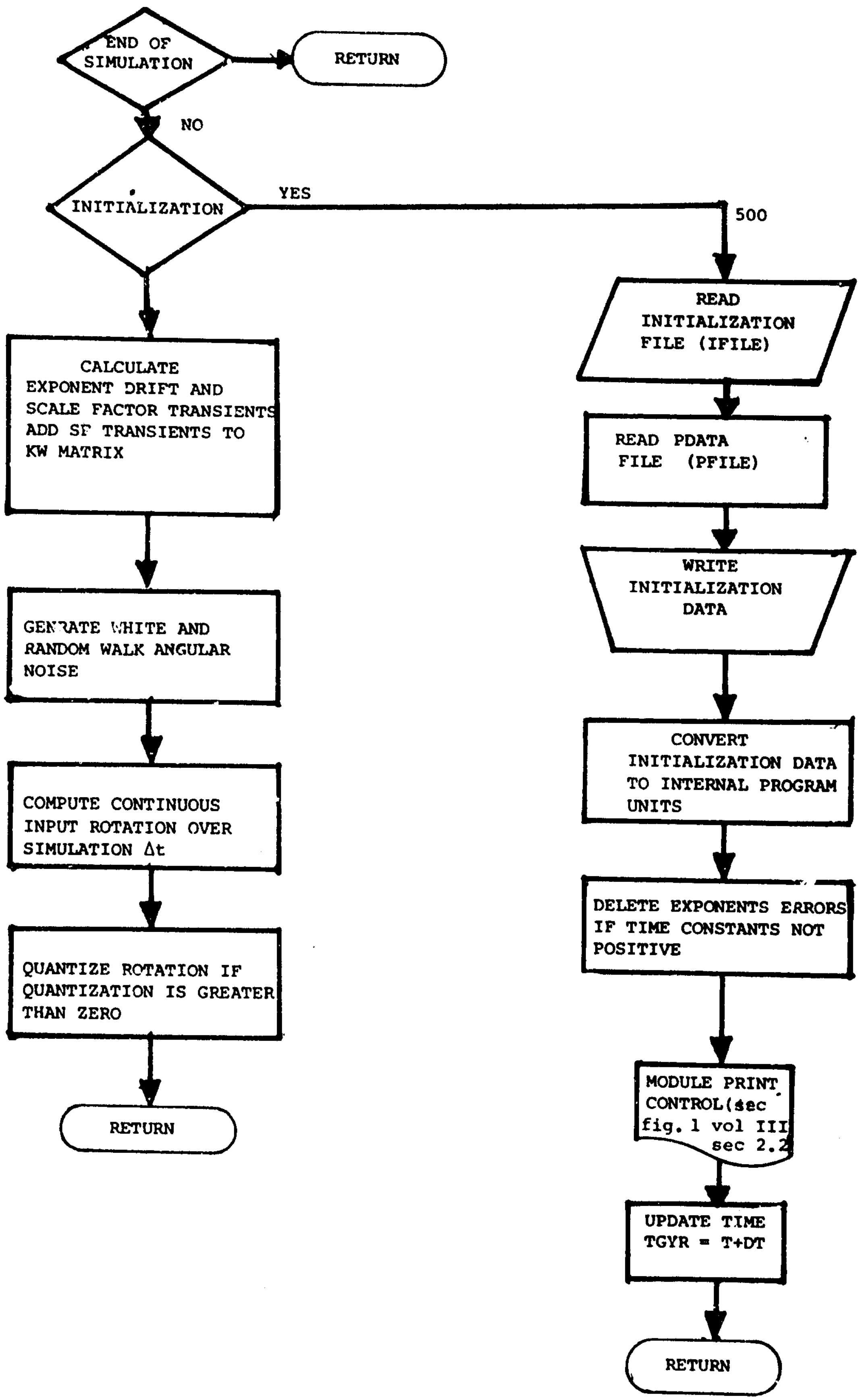

Figure 1. Laser Gyro Module Flow Diagram

Index	Variable	Default Value	Units	Description
1	DT	0.01	s	module operating time step
2	PRNTSW	1.0	logical	point switch (0 - no print) otherwise print)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	---	not used
6	SPARE2	0.0	---	not used
7	MODPDT	6.0	s	module print interval
8	DB(1)	0.7	deg/hr	gyro bias
9	DB(2)	0.4	deg/hr	
10	DB(3)	-0.6	deg/hr	
11	DTA(1)	0.06	deg/hr	transient drift amplitude
12	DTA(2)	-0.03	deg/hr	
13	DTA(3)	-0.05	deg/hr	
14	DTC(1)	45.0	min	transient drift time constant
15	DTC(2)	45.0	min	
16	DTC(3)	45.0	min	
17	SFTA(1)	0.8	ppm	transient scale factor amplitude
18	SFTA(2)	-0.7	ppm	
19	SFTA(3)	1.0	ppm	
20	SFTC(1)	45.0	min	transient scale factor time constant
21	SFTC(2)	45.0	min	
22	SFTC(3)	45.0	min	
23	KW(1)	1.0	ppm	gyro SF and gyro Input Axis (IA) misalignment matrix ● main diagonal = SF bias (ppm) ● off diagonal = input axis misalignment (microrad; ppm)
24	KW(2)	20.0	ppm	
25	KW(3)	50.0	ppm	
26	KW(4)	40.0	ppm	
27	KW(5)	1.5	ppm	
28	KW(6)	30.0	ppm	
29	KW(7)	-30.0	ppm	
30	KW(8)	-40.0	ppm	
31	KW(9)	-1.	ppm	
32	STDWN	0.0	sec	angular white noise std
33	STDRW	0.002	deg/$\sqrt{hr}$	random walk magnitude
34	Q	1.6	arcsec	angular quantization

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	---	not used

(4) Call-Line Data

$$\overbrace{\text{T, IENDF, WBB, WDOT, ABB,}}^{\text{INPUT}}$$

$$\underbrace{\text{DTHETA)}}_{\text{OUTPUT}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
$\overline{\text{WBB}}$	rad/s	REAL	angular rate vector of body wrt inertial space in body frame with vibration included
$\overline{\text{WDOT}}$	---	REAL	not used
$\overline{\text{ABB}}$	---	REAL	not used

(b) Call-Line Output

Variable	Units	Data Type	Description
$\overline{\text{DTHETA}}$	rad	REAL	quantized integral of angular rate about X, Y, Z gyro input axes over the computation cycle, plus errors

(5) Formulation

(a) Initialization Function

The switch INITSW is initialized to zero in DATA statement. The following functions are performed on the first pass, if INITSW = 0 and IENDF = 0 and the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization fiel (PFILE).
- Set OFILE = XFILE.
- Convert the initialization data to internal program units. Internal units are: radians, feet, and seconds.

$$\overline{DB} = \overline{DB} * DTR/3600$$

$$\overline{DTA} = \overline{DTA} * DTR/3600$$

$$\overline{DTC} = \overline{DTC} * 60$$

$$\overline{SFTA} = \overline{SFTA} * 1.E-6$$

$$\overline{SFTC} = \overline{SFTC} * 60$$

$$\overset{*}{KW} = \overset{*}{KW} * 1.E-6$$

$$Q = Q * DTR/3600$$

where DTR equals degrees-to-radians conversion factor.

- Compute the random walk amplitude coefficient KRW.

  KRW = (STDRW * DTR) * SQRT (DT/3600)

- Compute the exponential drift and exponential scale factor coefficients.

$$\overline{EXPD} = EXP(-DT/\overline{DTC})$$

$$\overline{EXPSF} = EXP(-DT/\overline{SFTC})$$

The vector $\overline{DTC}$ contains the time constants of the exponential drifts and $\overline{SFTC}$ contains the time constants of the exponential scale factors. The elements of $\overline{EXPD}$ and $\overline{EXPSF}$ are set to zero if the corresponding time constants are non-positive. The difference equation for each of the terms is of the form

$$D_T(n+1) = D_T(n) * EXPD$$

$$S_T(n+1) = S_T(n) * EXPSF$$

Thus, the individual transients are set to zero if the associated time constant is not positive. This logic allows the user to delete any of the transient terms by setting the amplitude and time constant to zero.

Initialization is now complete, INITSW is set to 1, and the simulation time is incremented.

$$TGYR = T + DT$$

Now, the routine returns to the main program

(b) General Functions

The following functions are performed every module operating cycle.

- Compute the transient drift and scale factor.

$$\overline{DTA} = \overline{DTA} * \overline{EXPD}$$

$$\overline{SFTA} = \overline{SFTA} * \overline{EXPSF}$$

The transient terms are set to zero if the magnitude becomes less than 1.E-10 to prevent underflow.

- Add the scale factor transient to the scale factor bias. Insert the sum on the main diagonal of the rate sensitive matrix KW^*.

$$KW(K) = SF(I) + SFTA(I), \text{ where } K = \text{diagonal elements.}$$

The off diagonal elements of KW^* contain the IA misalignment.

- Compute the white noise and angular random walk.

White Angle Noise

$$ANGWN(I) = STDWN * GAUSS(0.0, 1.0)$$

where

STDWN = specified white noise standard deviation

GAUSS(0.0, 1.0) zero mean, unit variance Gaussian random number

Angular Random Walk

$$ANGRW(I) = KRW * GAUSS\ (0.0, 1.0)$$

where

KRW = random walk amplitude coefficient computed in the initialization section

GAUSS(0.0, 1.0) = zero mean, unit variance Gaussian randomm number

- Compute the gyro output angular velocity.

$$\overline{W} = (\overset{*}{I} + \overset{*}{KW})\ \overline{WBB} + \overline{DB} + \overline{DTA}$$

where

$\overline{WBB}$ = angular velocity of the body in body coordinates

$\overline{DB}$ = gyro bias

$\overline{DTA}$ = gyro drift transient

$\overset{*}{KW}$ = angular rate sensitive matrix

$\overset{*}{I}$ = identity matrix

- Compute the continuous output rotation angle over the simulation time step including the white and random walk angular noise.

$$\overline{ANG} = \overline{ANG} + \overline{W} * DT + \overline{ANGWN} + \overline{ANGRW}$$

The previous value of the angle $\overline{ANG}$ is the quantization residual.

- Compute the quantized indicated rotation over the simulation time step, QANG, and the new angle residual, ANG.

$$NP = \text{Integer} (\overline{ANG}/Q)$$

$$QANG = NP * Q$$

$$\overline{ANG} = \overline{ANG} - QANG - \overline{ANGWH}$$

The quantized rotation QANG is set equal to the continuous rotation angle if the quantization amplitude is zero.

- Sum the quantized rotations to form the incremental gyro output angle DTHETA. The output angle is accumulated until the platform attitude matrix is update - then the navigation routines reset DTHETA to zero.

$$\overline{DTHETA} = \overline{DTHETA} \_ QANG$$

- The DTHETA is printed

- The simulation time is incremented

$$TGYR = T + DT$$

and the subroutine returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass the title "LASER GYRO INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for print control logic. The printed output is, as follows

(7) <u>Subroutines Called</u> (See Section 2.3.15)

MXV = matrix by vector product.

GAUSS(0.0, 1.0) = zero mean, unit variance Gaussian random number generator

2.3.6 ACCELEROMETER MODULE (ACCEL)

(1) General Description

The ACCEL module simulates the dynamics of a pendulous SDF floated accelerometer. The kinematics of the instrument can be selectively modeled using either a second-order or a first-order differential equation, or a "performance" model. The following accelerometer characteristics are modeled:

- Acceleration sensitivities (both g and g-squared).
- Anisoinertia.
- Output-axis coupling.
- Scale-factor errors (both positive and negative).
- Scale-factor-error acceleration sensitivity (both positive and negative).
- Lever-arm effects (including accompanying bias).
- Quantization level.
- Exponentially correlated random bias.

Although the parameters characterizing these effects are entirely selectable, the nominal (default) parameters specified in the initialization data file correspond to the Systron Donner 4841 accelerometer with its analog rebalance loop.

(2) Accelerometer Module Flow Diagram

The general flow logic of the ACCEL module is illustrated in Figure 1.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 50

FORTRAN format: I5, 1X, F20.10

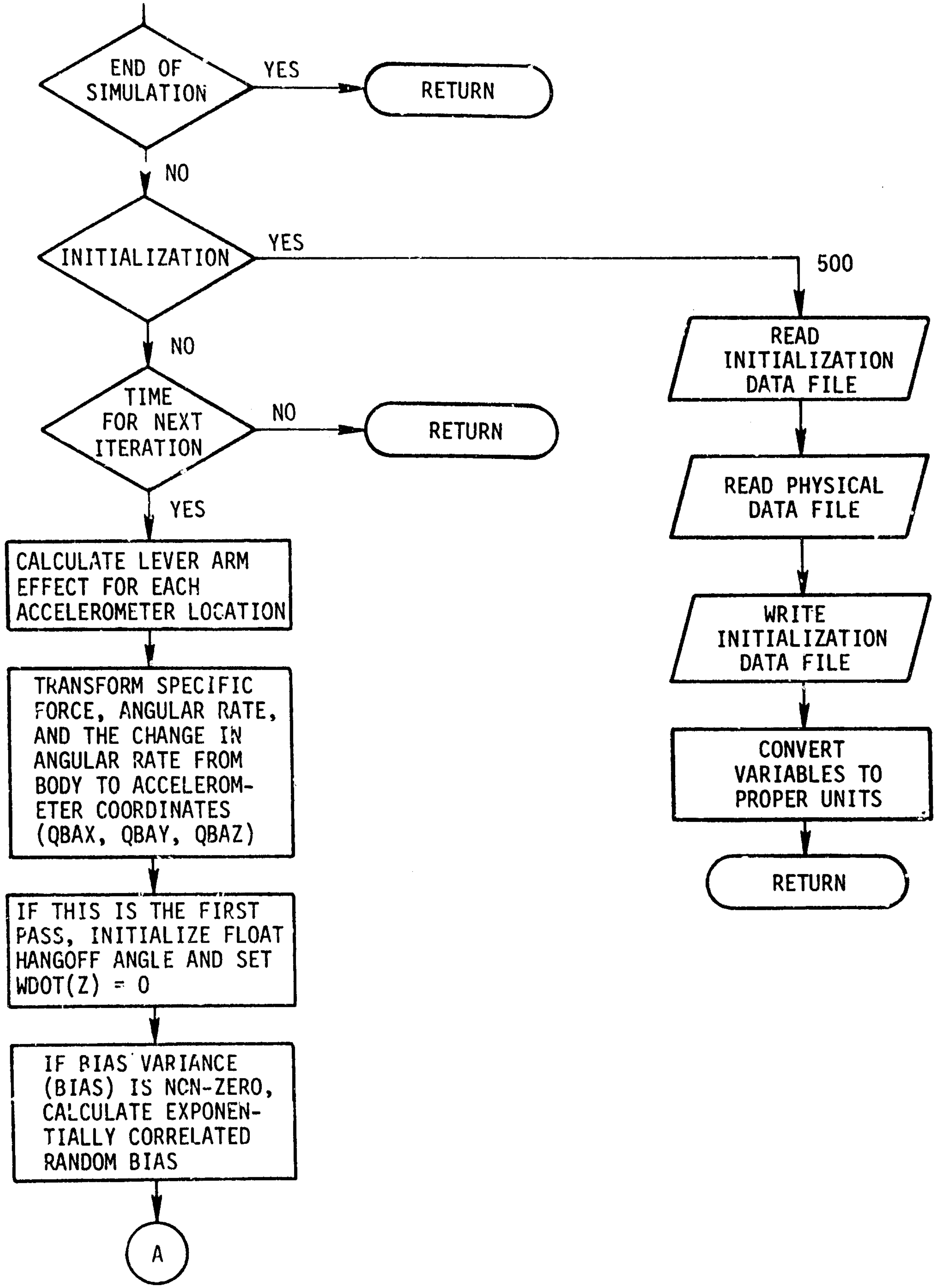

Figure 1. Accelerometer module. (Sheet 1 of 2)

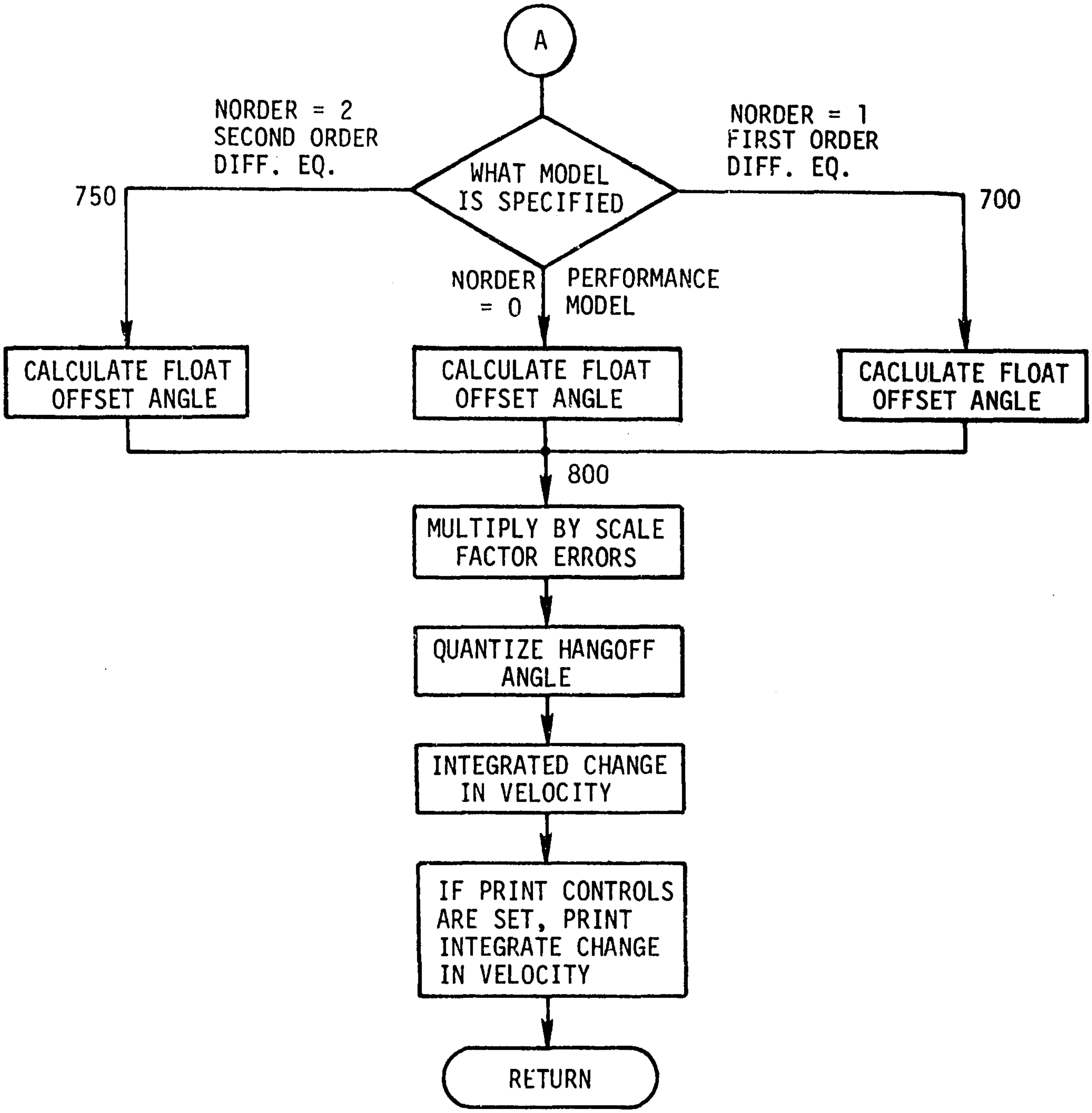

Figure 1. Accelerometer module. (Sheet 2 of 2)

Index	Variable	Default Value	Units	Description
1	DT	0.01	s	module operating cycle time
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	---	not used
6	SPARE2	0.0	---	not used
7	CO	25000.	dyn-cm/rad/s	output-axis damping coefficients
8-12	--------	----	---	not used
13	QBAX(1)	1.0	unity	transformation matrix from body coordinates to X-accelerometer (input, output, pendulous) coordinates
14	QBAX(2)	0.0	unity	
15	QBAX(3)	0.0	unity	
16	QBAX(4)	0.0	unity	
17	QBAX(5)	0.0	unity	
18	QBAX(6)	1.0	unity	
19	QBAX(7)	0.0	unity	
20	QBAX(8)	-1.0	unity	
21	QBAX(9)	0.0	unity	
22	QBAY(1)	0.0	unity	transformation matrix from body coordinates to Y-accelerometer (input, output, pendulous) coordinates
23	QBAY(2)	1.0	unity	
24	QBAY(3)	0.0	unity	
25	QBAY(4)	1.0	unity	
26	QBAY(5)	0.0	unity	

Index	Variable	Default Value	Units	Description
27	QBAY(6)	0.0	unity	
28	QBAY(7)	0.0	unity	
29	QBAY(8)	0.0	unity	
30	QBAY(9)	-1.0	unity	
31	QBAZ(1)	0.0	unity	transformation matrix from body coordinates to Z-accelerometer (input, output, pendulous) coordinates
32	QBAZ(2)	0.0	unity	
33	QBAZ(3)	1.0	unity	
34	QBAZ(4)	1.0	unity	
35	QBAZ(5)	0.0	unity	
36	QBAZ(6)	0.0	unity	
37	QBAZ(7)	0.0	unity	
38	QBAZ(8)	1.0	unity	
39	QBAZ(9)	0.0	unity	
40	MRC	0.7	gm-cm	pendulosity parameter
41	---	---	---	not used
42	QUANT	1.0	ft/s/pulse	quantization level of increment change in velocity
43	BIAS(1)	900.0	g	X accelerometer bias
44	BIAS(2)	900.0	g	Y accelerometer bias
45	BIAS(3)	900.0	g	Z accelerometer bias
46	K	100000000.	$gm\text{-}cm^2/s^2$	elastic restraint of rebalance loop
47	I	5.0	$gm\text{-}cm^2$	output-axis moment of inertia
48	DELI	0.0	$gm\text{-}cm^2$	difference of moment of inertia between pendulous and input axes

Index	Variable	Default Value	Units	Description
49-50				
51	KO(1)	0.0	μg/g	coefficient of output-axis acceleration sensitivity X accelerometer
52	KO(2)	0.0	μg/g	Y
53	KO(3)	0.0	μg/g	Z
54	KP(1)	0.0	μg/g	coefficient of output-axis acceleration sensitivity X accelerometer
55	KP(2)	0.0	μg/g	Y
56	KP(3)	0.0	μg/g	Z
57	KII(1)	0.8	$\mu g/g^2$	coefficients of input-axis acceleration squared sensitivity X accelerometer
58	KII(2)	0.8	μg/g	Y
59	KII(3)	0.8	$\mu g/g^2$	Z
60	KPP(1)	0.0	$\mu g/g^2$	coefficients of pendulous-axis acceleration-squared sensitivity X accelerometer
61	KPP(2)	0.0	$\mu g/g^2$	Y
62	KPP(3)	0.0	$\mu g/g^2$	Z
63	KIO(1)	0.0	$\mu g/g^2$	coefficient of input by output-axis acceleration sensitivity X accelerometer
64	KIO(2)	0.0	$\mu g/g^2$	Y
65	KIO(3)	0.0	$\mu g/g^2$	Z
66	KIP(1)	0.0	$\mu g/g^2$	coefficient of input-by pendulous-axis acceleration sensitivity X accelerometer
67	KIP(2)	0.0	$\mu g/g^2$	Y
68	KIP(3)	0.0	$\mu g/g^2$	Z
69	KOP(1)	0.0	$\mu g/g^2$	coefficient of output-by pendulous-axis acceleration sensitivity X accelerometer
70	KOP(2)	0.0	$\mu g/g^2$	Y
71	KOP(3)	0.0	$\mu g/g^2$	Z

Index	Variable	Default Value	Units	Description
72	BIASV(1)	0.0	g^2	exponentially correlated random bias variances for X accelerometer
73	BIASV(2)	0.0	g^2	exponentially correlated random bias variances for Y accelerometer
74	BIASV(3)	0.0	g^2	exponentially correlated random bias variances for Z accelerometer
75	SFPO(1)	10.0	ppm	positive scale-factor errors for X accelerometer
76	SFPO(2)	10.0	ppm	positive scale-factor errors for Y accelerometer
77	SFPO(3)	10.0	ppm	positive scale-factor errors for Z accelerometer
78	SFMO(1)	10.0	ppm	negative scale-factor errors for X accelerometer
79	SFMO(2)	10.0	ppm	negative scale-factor errors for Y accelerometer
80	SFMO(3)	10.0	ppm	negative scale-factor errors for Z accelerometer
81	MODPDT	6.0	s	module print interval
82	ORDER	0.0	logical	order of differential equation model 0 = performance model 1 = first-order diff. equation 2 = second-order diff. equation
83	SFP1(1)	0.0	ppm/ rad/s	positive rate sensitivity of scale-factor errors X accelerometer
84	SFP1(2)	0.0	ppm/ rad/s	positive rate sensitivity of scale-factor errors Y accelerometer
85	SFP1(3)	0.0	ppm/ rad/s	positive rate sensitivity of scale-factor errors Z accelerometer
86	SFM1(1)	0.0	ppm/ rad/s	negative rate sensitivity of scale-factor errors X accelerometer
87	SFM1(2)	0.0	ppm/ rad/s	negative rate sensitivity of scale-factor errors Y accelerometer
88	SFM1(3)	0.0	ppm/ rad/s	negative rate sensitivity of scale-factor errors Z accelerometer

Index	Variables	Default Value	Units	Description
89	RX(1)	1.0	ft	position vector in body coordinates (vehicle cg to X-accelerometer origin)
90	RX(2)	0.0	ft	
91	RX(3)	0.0	ft	
92	RY(1)	0.9	ft	position vector in body coordinates (vehicle cg to Y-accelerometer origin)
93	RY(2)	0.2	ft	
94	RY(3)	0.0	ft	
95	RZ(1)	0.9	ft	position vector in body coordinates (vehicle cg to Z-accelerometer origin)
96	RZ(2)	0.0	ft	
97	RZ(3)	0.2	ft	

(b) <u>Common Initialization File (PFILE)</u>

FORTRAN unit number: 7

FORTRAN format: I5, 1X, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing cycle time
5	PBUF(1)	0.0	---	not used

(4) <u>Call-Line Data</u>

$$\overbrace{(T,\ IENDF,\ \overline{ABB},\ \overline{WBB},\ \overline{WDOT},}^{\text{Input}}$$

$$\underbrace{\overline{DV}}_{\text{Output}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
$\overline{ABB}$	ft/s^2	REAL	specific-force vector in body frame with vibration
$\overline{WBB}$	rad/s	REAL	angular rate vector of body wrt inertial space in body frame with vibration included
$\overline{WDOT}$	rad/s^2	REAL	angular acceleration vector of body wrt inertial space body frame with vibration included

(b) Call-Line Output

Variable	Units	Data Type	Description
$\overline{DV}$	ft/s	REAL	quantized integral of the vehicle specific-force vector along the X, Y, Z accelerometer input axes

(5) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE and NORDER = ORDER
- Convert initialization data to computational units.

$$\overline{SPO} = \overline{SFGPO} * 1.E - 6$$

$$\overline{SP1} = \overline{SFGP1} * 1.E - 6/G$$

$$\overline{SMO} = \overline{SFGMO} * 1.E - 6$$

$$\overline{SM1} = \overline{SFGM1} * 1.E - 6/G$$

$$AQUANT = QUANT * MRC/K/30.48/DT$$

$$EX = EXP(-DT/40)$$

$$\overline{BIASA} = (BIASV * (1 - EX^2))^{1/2} * 1.E - 6 * G$$

$$\overline{BIAS} = BIAS * 1.E - 6 * G$$

In addition to conversion to computational units, the following variables are tested for a zero value and the computation bypass switches K11 and K12, which are initialized to zero in the DATA statement, are reset

$$\text{if } \overline{BIASV} \neq 0, \text{ set } K11 = 1$$

Set

$$\overline{KO} = \overline{KO}*1.E - 6 \text{ and if } \overline{KO} \neq 0, \text{ set } K12 = 1,$$

$$\overline{KP} = \overline{KP}*1.E - 6 \text{ and if } \overline{KP} \neq 0, \text{ set } K12 = 1,$$

$$\overline{KII} = \overline{KII}*1.E - 6/G \text{ and if } \overline{KII} \neq 0, \text{ set } K12 = 1,$$

$$\overline{KIO} = \overline{KIO}*1.E - 6/G \text{ and if } \overline{KIO} \neq 0, \text{ set } K12 = 1,$$

$$\overline{KIP} = \overline{KIP}*1.E - 6/G \text{ and if } \overline{KIP} \neq 0, \text{ set } K12 = 1,$$

$$\overline{KOP} = \overline{KOP}*1.E - 6/G \text{ and if } \overline{KOP} \neq 0, \text{ set } K12 = 1,$$

$$\overline{KPP} = \overline{KPP}*1.E - 6/G \text{ and if } \overline{KPP} \neq 0, \text{ set } K12 = 1,$$

where 1.E - 6 is the factor for converting μgs and G is the nominal gravity.

Continuing,

$$\begin{aligned}
MRC &= MRC*7.23E - 5 \\
CO &= CO*2.37E - 6 \\
DELI &= DELI*2.37E - 6 \\
K &= K*2.37E - 6 \\
I &= I*2.37E - 6 \\
DTI &= DT/I
\end{aligned}$$

The factor 2.37E - 6 is the conversion factor from gm-cm^2 to lb-ft^2 and the factor 7.23E - 5 is for conversion of gm-cm to lb-ft.

Now set INITSW = 1, increment the simulation time

$$TACC = T + DT$$

and return the subroutine to the main program.

(b) General Function

The following functions are performed every module operating cycle. The index K10 is incremented, K10 = K10 + 1 from an initial value of zero.

The quantized integral of specific force vector $\overline{DV}$, for the X, Y, Z accelerometer is calculated in turn.

First for the X-accelerometer (when I1 = 1) acceleration from the lever-arm effect is calculated from the components of angular rate and the accelerometer position vector. This acceleration plus the input acceleration become the total acceleration sensed by the X accelerometer.

$$\overline{ABC} = \underbrace{\overline{ABB}}_{\text{input}} + \underbrace{(\overline{WBB} \times (\overline{WBB} \times \overline{RX})) + \overline{WDOT} \times \overline{RX}}_{\text{lever arm}}$$

where in the mnemonics of the program, initially

$$\overline{CROSS1} = \overline{WBB} \times \overline{RX}$$

and

$$\overline{CROSS2} = \overline{WBB} \times \overline{CROSS1}$$

and now redefining

$$\overline{CROSS1} = \overline{WDOT} \times \overline{RX}$$

The coordinates of the specific force, angular rate, and angular acceleration are transformed from (XYZ) body coordinates to (IOP) accelerometer coordinates

$$\overline{ABA} = \overset{*}{QBAX} * \overline{ABC}$$

$$\overline{WBA} = \overset{*}{QBAX} * \overline{WBB}$$

$$\overline{WDOTA} = \overset{*}{QBAX} * \overline{WDOT}$$

Similarily, for the Y-accelerometer the sensed acceleration is calculated

$$\overline{ABC} = \underbrace{\overline{ABB}}_{\text{input}} + \underbrace{(\overline{WBB} \times (\overline{WBB} \times \overline{RY})) + \overline{WDOT} \times \overline{RY}}_{\text{lever arm}}$$

and the specific force, angular rate, and angular acceleration are transformed to (IOP) coordinates

$$\overline{ABA} = \overset{*}{QBAY} * \overline{ABC}$$

$$\overline{WBA} = \overset{*}{QBAY} * \overline{WBB}$$

$$\overline{WDOTA} = \overset{*}{QBAY} * \overline{WDOT}$$

Finally, for the Z accelerometer

$$\overline{ABC} = \underbrace{\overline{ABB}}_{\text{input}} + \underbrace{(\overline{WBB} \times (\overline{WBB} \times \overline{RZ})) + \overline{WDOT} \times \overline{RZ}}_{\text{lever arm}}$$

and

$$\overline{ABA} = \overset{*}{QBAZ} * \overline{ABC}$$

$$\overline{WBA} = \overset{*}{QBAZ} * \overline{WBB}$$

$$\overline{WDOTA} = \overset{*}{QBAZ} * \overline{WDOT}$$

Now for accelerometers X, Y, and Z, when I1 = 1, 2, and 3, respectively, the following terms are calculated.

If, K10 = 1, which occurs on the first-pass filter after initialization, then set

$$THETA(I1) = (MRC/K) * ABA(1)$$

and $$WDOTA(2) = 0.0$$

The output axis value bias levels are now calculated. If the bias variance is zero, K11 = 0, and the random bias is set to the bias level,

$$BIA = BIAS(I1)$$

Otherwise, a Gaussian random number with zero mean and standard deviation, BI, where BI = BIASA(I1) is generated and the exponentially correlated random bias, BIA, is calculated, as follows. The value of ABIAS for the particular accelerometer, I1, from the previous pass (on the first pass after the initialization pass, ABIAS = (0.0, 0.0, 0.0)) is multiplied by the exponential function, EX, and added to the random number from the GAUSS function.

$$ABIAS(I1) = ABIAS(I1) * EX + GAUSS(AN,BI)$$

where

$$EX = EXP(-DT/40).$$

Finally, the exponentially correlated random bias for the I1 accelerometer is

$$BIA = BIAS(I1) + ABIAS(I))$$

The dynamic equations of the pendulous single-degree-of-freedom floated accelerometer are solved for the output-axis float offset angle, THN, for any one of three approximate forms:

Performance model, when NORDER = 0.

First-order model, when NORDER = 1.

Second-order model, when NORDER = 2.

See Volume II, Section 6 for a theoretical development of the dynamic equations.

A quantity DEN is defined

$$DEN = K + DELI*(WBA(3)^2 - WBA(1)^2) - MRC*ABH(3)$$

where the subscripts are

1 = input axis
2 = output axis
3 = pendulous axis

The quantity DEN times the float offset angle is a torque term. An additional torque term, M1, is defined as

$$M1 = MRC*(ABA(1) - BIA) + DELI * WBA(3)*WBA(1) - I*WDOTA(2)$$

Additional torque terms in the numerator, denoted by M2, are computed for acceleration and acceleration-squared sensitive effects if any of the coefficients are nonzero, i.e., when K12 = 1.

M2 = MRC*(-KP(I1)*ABA(3) - KO(I1)*ABA(2)

- KPP(I1)*ABA(3)*ABA(3) - KII(I1)*ABA(1)*ABA(1)

- KIP(I1)*ABA(1)*ABA(3) - KIO(I1)*ABA(1)*ABA(2)

- KOP(I1)*ABA(2)*ABA(3))

Otherwise, M2 = 0.0

If NORDER = 0, the performance model equation is solved for the float offset angle, THN

THN = (M1 + M2)/DEN

If NORDER = 1, the first-order model equation is solved for THN.

THN = THETA(I1) + DT*(-DEN*THETA(I1) + M1 + M2)/CO

Finally, if NORDER = 2, the second-order model equation is solved for THN

THN = THETA(I1) + DT*THDOT(I1)

where THDOT(I1) is calculated on the previous pass
Now THDOT(I1) is calculated for the next pass from

THDOT(I1) = THDOT(I1) + (-CO*THDOT(I1) - DEN*THETA(I1) + M1 + M2)*DTI

The quantized output is now calculated. The float offset angle, THN, is added to the residual angle, TTHET, from the previous pass

TTHET(I1) = TTHET(I1) + THN

Scale-factor error corrections are computed for positive accelerations if the residual angle, TTHET(I1), is greater than or equal to zero from

$$T1 = SP0(I1) + 1 + SP1(I1)*THN*K/MRC$$

or if it is less than zero for negative acceleration from

$$T1 = SM0(I1) + 1 + SM1(I1)*THN*K/MRC$$

The sum of the float offset angle and the residual angle are corrected for the scale-factor errors by multiplying by T1, and this quantity is divided by the quantization level, AQUANT. The truncation of this quantity

$$ITHET = TTHET(I1)/AQUANT/TI$$

results in the quantized float offset angle, i.e. the count of quantization pulses contained in it.

A new residual angle, i.e. the portion dropped off at truncation, is calculated from

$$TT = FLOAT(ITNET)$$

and $$TTHET(I1) = TTHET(I1) - TT*TI*AQUANT$$

and the float offset angle is saved for the calculations of the next pass

$$THETA(I1) = THN$$

Now, the desired output from the ACCEL module, $\overline{DV}$, is obtained for the X, Y, and Z accelerometers (I1 = 1, 2, or 3, respectively) by integrating

$$DV(I1) = DV(I1) + TT*AQUANT/MRC*K*DT$$

where

$$TT = FLOAT(ITHET) \text{ (maintains real variable arithmetic)}$$

This summation is carried out to equalize the cycle intervals for the hardware and software. $\overline{DV}$ is initialized in the RDR module (the hardware-software interface) each RDR cycle.

The quantized integrated change in velocity is printed, the simulation time is incremented

$$TACC = T + DT$$

and the subroutine returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass the title "ACCELEROMETER INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW 1. See Section 2.2 for print control logic. The printed output is, as follows

Variable	Units	Description
$\overline{DV}$	ft/s	quantized integral of specific force the X, Y, and Z accelerometer input axes over the computation cycle plus errors

(7) Subroutines Called (See Section 2.3.15)

MXV = matrix by vector product

2.3.7 ALTIMETER MODULE (ALTI)

(1) General Description

The ALTI module is designed to simulate the output characteristics of a barometric altimeter, incorporating errors in its air-transducer assembly and uncertainties caused by nonstandard-atmosphere and Pitot-static errors. The integration of this simulated instrument into the INSS gives rise to an indicated altitude which has the following characteristics:

- Long-term stability of the barometric altimeter.
- A resultant frequency response that is much faster than that of a pure barometric altimeter, and which attenuates the effects of barometric noise.
- The relative insensitivity to low-frequency accelerometer errors resolved into the vertical channel.

Essentially, the indicated altimeter output is the sum of the "true" vehicle altitude (passed from the trajectory module) and user-characterized exponentially correlated random bias and random noise. Although a perfect altimeter can be simulated by this module through user-specification of an on/off switch (NOISSW = 1) in the attendant initialization data file, the nominal (default) initialization parameters correspond to the instrument projected for use on the Space Shuttle.

(2) Altimeter Module Flow Diagram

The general flow logic of the ALTI module is shown in Figure 1.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 60

FORTRAN format: I5, F20.10

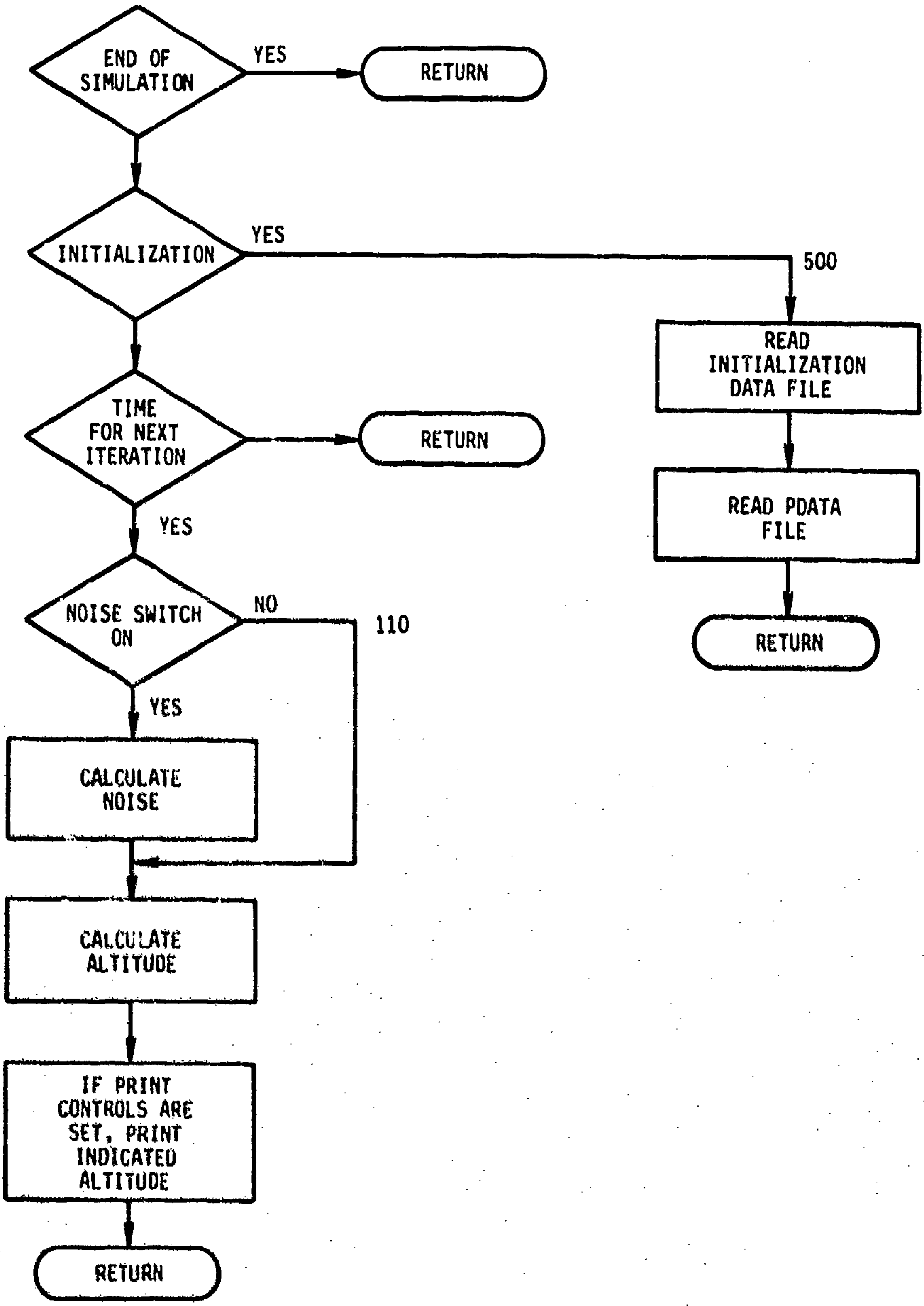

Figure 1. Altimeter module.

Index	Variable	Default Value	Units	Description
1	DT	0.02	s	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise pring)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	MODPDT	6.0	s	module print interval
6	NOISSW	0.0	logical	noise generator switch
7	TC	40.0	s	exponentially correlated random noise time constant
8	U0	0.000000000000040	$1/ft^2$	variance coefficients relating altitude to bias uncertainty
9	U1	0.000000023	$(s^2/ft)^2$	variance coefficients relating velocity to bias uncertainty
10	U2	17000.	ft^2	variance coefficients relating bias uncertainty constant
11	U3	0.000000000625	---	variance coefficients relating altitude to random bias uncertainty
12	U4	97.0	ft^2	variance coefficients relating random-vibration uncertainty (constant random vibration)

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	---	not used

(4) Call-Line Data

$$\overbrace{(T,\ IENDF,\ ALT,\ \bar{V}}^{\text{INPUT}}\ \underbrace{ALTO}_{\text{OUTPUT}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
ALT	ft	REAL	"true" altitude (from trajectory module)
$\tilde{V}$	ft/s	REAL	"true" velocity vector in body frame (from trajectory module)

(b) Call-Line Output

Variable	Units	Data Type	Description
ALTO	ft	REAL	indicated barometric altitude

(5) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in an DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0 and if the module operating cycle time has elapsed.

Read and print initialization file (FILE). Read common initialization file (PFILE). Set

$$\begin{aligned} OFILE &= XFILE \\ ALTO &= ALT \\ EX &= EXP\ (-DT/TC) \\ E2X1 &= 1 - EX^2 \\ INITSW &= 1 \end{aligned}$$

The simulation time is incremented

$$TALT = T + DT$$

and the subroutine returns to the main program.

(b) General Function

The following functions are performed every module operating cycle.

If the noise generator switch is, NOISSW = 0, the indicated altitude, ALTO, is equal to the true altitude, ALT. Otherwise, the barometric altitude with noise added is calculated.

The model chosen to replicate the output of a barometric altimeter is fundamentally stochastic. As such, five statistical parameters (variances) specified in the attendant data file are employed to generate random errors. These errors are then summed with the "true" altitude (ALT) passed over from the trajectory module to simulate an output measurement. The five input statistics represent the variances which characterize: bias uncertainties relative to altitude (U_0) and velocity (U_1), constant bias uncertainty (U_2), random-vibration uncertainty relative to altitude (U_3), and constant random-vibration uncertainty (U_4). These variances are subsequently combined by the algorithm to generate both exponentially correlated random bias and Gaussian noise over each cycle.

A first term (the random noise), a normally distributed random number with a mean, AM = 0.0, and a standard deviation given by

$$SIGR = (U3*ALT^2 + U4)^{1/2}$$

is computed using the GAUSS function.

Another term (the random bais), a normally distributed random number with a mean, AM = 0.0, and a standard deviation given by

$$SIGN = \left[(U0*(ALTA2)^2 + U1*(VI2)^2 + U2) *E2X1\right]^{1/2}$$

where

$$ALTA2 = ALT2$$
$$VI2 = V(1)^2 + V(2)^2 + V(3)^2$$

is obtained by the use of the GAUSS subroutine. An exponentially correlat d noise term, X, is compute from the random bias term,

$$X = EX*X + GAUSS(AM,SIGN)$$

where X on the right-hand side of the equation is the value of X on the previous pass (initially zero).

The indicated barometric altitude, ALTO, is computed as the sum of the true altutude, the random bias and the random noise term

$$ALTO = ALT + X + GAUSS(AM, SIGR)$$

The indicated altitude, ALTO, is printed and the simulation time is incremented

$$TALT = T + DT$$

Now the subroutine returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6.

On the initialization pass the title "ALTIMETER INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRINTSW $\geq$ 1. See Section 2.2 for print control logic. The printed output is, as follows

Variable	Units	Description
ALTO	ft	indicated barometric altitude

(7) Subroutines Called (See Section 2.3.15)

GAUSS(AM,SD) = Gaussian random number generator function with mean, AM, and standard deviation, SD.

2.3.8 HARDWARE/SOFTWARE INTERFACE MODULE (RDR)

(1) General Description

Functioning as a buffer between the fast and slow cycles of the INSS executive (sequencer), this module effectively simulates the action of the hardware/software interface. After reading the accumulated incremental angle and velocity components, the interface module passes the appropriate values to the compensation algorithms. The face values previously accumulated are then reset to zero and returned to the gyro- and accelerometer-module accumulation registers.

(2) Hardware/Software Interface Module Flow Diagram

The general flow logic of the RDR module is shown in Figure 1.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 65

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.02	s	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	---	not used
6	MODPDT	6.0	s	module print interval

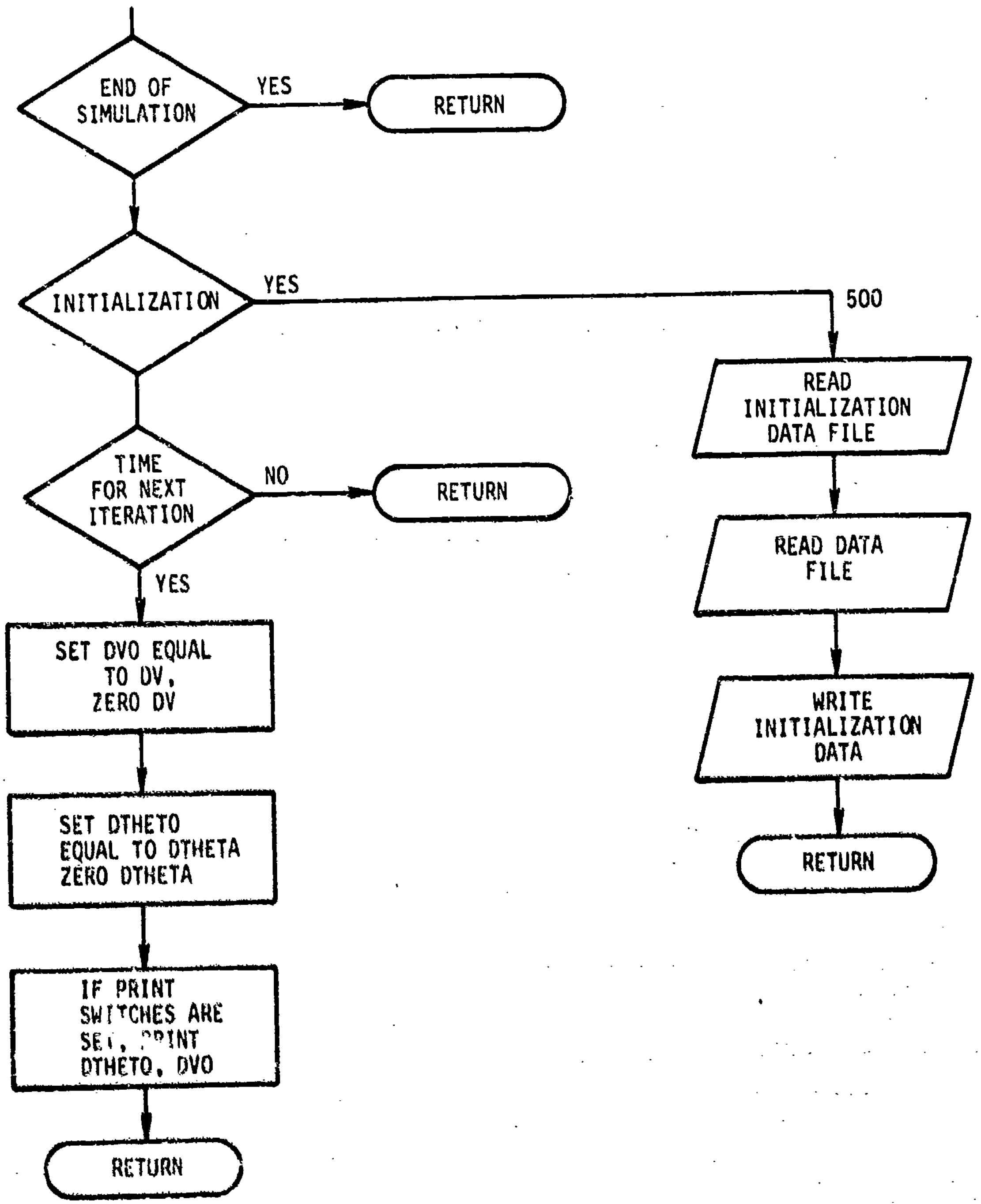

Figure 1. Hardware/software interface module flow diagram.

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	---	not used

(4) Call-Line Data

$$\overbrace{(T,\ IENDF,\ \underbrace{\overline{DTHETA},\ \overline{DV},}_{\text{In/Out}}}^{\text{Input}}$$

$$\underbrace{\overline{DTHETO},\ \overline{DVO})}_{\text{Output}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
$\overline{DTHETA}$	rad	REAL	quantized integral of angular rate about X, Y, Z gyro input axes over the computation cycle plus errors
$\overline{DV}$	ft/s	REAL	quantized integral of the vehicle specific force vector along the X, Y, Z accelerometer input axes

(b) Call-Line Output

Variable	Units	Data Type	Description
$\overline{\text{DTHETO}}$	rad	REAL	quantized integral of angular rate about X, Y, Z gyro input axes generated each fast cycle, and accumulated over the slow cycle
$\overline{\text{DVO}}$	ft/s	REAL	quantized integral of the vehicle specific force vector along the X, Y, Z accelerometer input axes generated each fast cycle, and accumulated over the slow cycle

(5) Formulation

(a) Initialization Function

The switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE and INITSW = 1.

The simulation time is incremental

$$\text{TRDR} = \text{T} + \text{DT}$$

and the subroutine returns to the main program.

(b) General Function

The following functions are performed every module operating cycle.

The input variables $\overline{DV}$ and $D\overline{THETA}$ are stored as the corresponding output variables, $\overline{DVO}$ and $D\overline{THETO}$, then the input variables are reset to zero (and returned to the accelerometer and gyro modules).

$$\overline{DVO} = \overline{DV}$$
$$\overline{DV} = 0.0$$
$$D\overline{THETO} = D\overline{THETA}$$
$$D\overline{THETA} = 0.0$$

The quantized integrals of angular rate, $D\overline{THETO}$, and specific force, $\overline{DVO}$ are printed and the simulation time is incremented

$$TRDR = T + DT$$

Now the subroutine returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass the title, "READER INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for print control logic. The printout is as follows

Variable	Units	Description
$\overline{DVO}$	ft/s	quantized integral of specific force along the X, Y, Z accelerometer input axes accumulated over slow cycle
$D\overline{THETO}$	rad	quantized integral of angular rate about the X, Y, Z gyro input axes accumulated over slow cycle

(7) Subroutines Called (See Section 2.3.15)

No subroutines are called.

2.3.9 ACCELEROMETER COMPENSATION MODULE (ACOMP)

(1) General Description

This software module computes the compensation for accelerometer bias, scale-factor errors, scale-factor errors with acceleration, the acceleration-squared terms along the input axis, accelerometer misalignments, anisoinertia, output-axis coupling, and lever-arm effects.

(2) Accelerometer Compensation Module Flow Diagram

Figure 1 depicts the general flow logic of the ACOMP module.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 67

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.02	s	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	---	not used
6	SPARE2	0.0	---	not used
7	DELI	0.0	$gm\text{-}cm^2$	difference of moments of inertia about pendulous and input axes

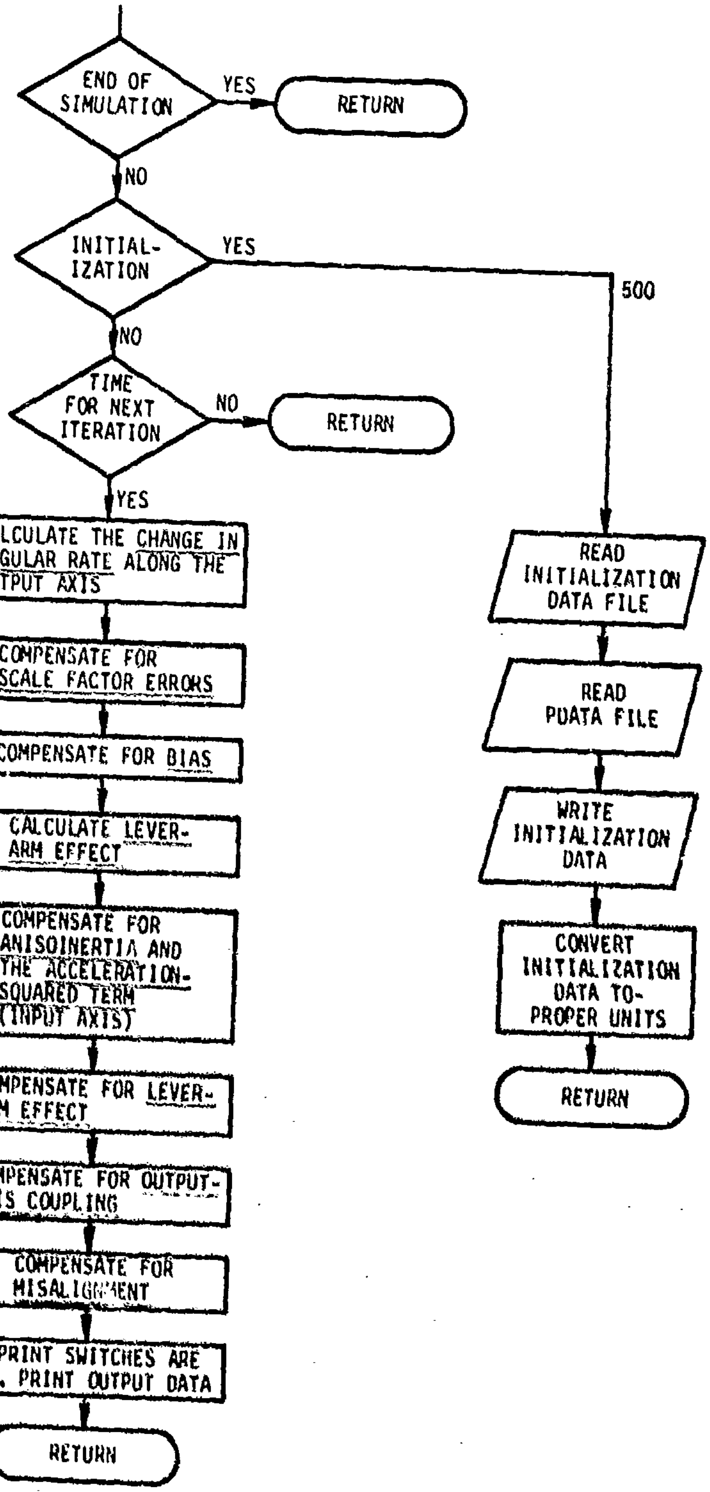

Figure 1. Accelerometer compensation module.

Index	Variable	Default Value	Unit	Description
8	QABX(1)	1.0	unity	transformation matrix from body coordinates to X-accelerometer (input, output, pendulous) coordinates
9	QABX(2)	0.0	unity	
10	QABX(3)	0.0	unity	
11	QABX(4)	0.0	unity	
12	QABX(5)	0.0	unity	
13	QABX(6)	1.0	unity	
14	QABX(7)	0.0	unity	
15	QABX(8)	-1.0	unity	
16	QABX(9)	0.0	unity	
17	SFPO(1)	10.0	ppm	X-accelerometers — positive scale-factor error
18	SFPO(2)	10.0	ppm	Y-accelerometer — positive scale-factor error
19	SFPO(3)	10.0	ppm	Z-accelerometer — positive scale-factor error
20	BIAS(1)	890.0	μg	X-accelerometer — bias
21	BIAS(2)	890.0	μg	Y-accelerometer — bias
22	BIAS(3)	890.0	μg	Z-accelerometer — bias
23	QABY(1)	0.0	unity	transformation matrix from body coordinates to Y-accelerometer (input, output, pendulous) coordinates
24	QABY(2)	1.0	unity	
25	QABY(3)	0.0	unity	
26	QABY(4)	1.0	unity	
27	QABY(5)	0.0	unity	
28	QABY(6)	0.0	unity	
29	QABY(7)	0.0	unity	
30	QABY(8)	0.0	unity	
31	QABY(9)	-1.0	unit	

Index	Variable	Default Value	Unit	Description
32	QABZ(1)	0.0	unity	transformation matrix from body coordinates to Z-accelerometer (input, output, pendulous) coordinates
33	QABZ(2)	0.0	unity	
34	QABZ(3)	1.0	unity	
35	QABZ(4)	1.0	unity	
36	QABZ(5)	0.0	unity	
37	QABZ(6)	0.0	unity	
38	QABZ(7)	0.0	unity	
39	QABZ(8)	1.0	unity	
40	QABZ(9)	0.0	unity	
41	MODPDT	6.0	s	module print interval
42	KII(1)	0.4	$\mu g/g^2$	X-accelerometer compensation input axis acceleration-squared coefficients
43	KII(2)	0.4	$\mu g/g^2$	Y-accelerometer
44	KII(3)	0.4	$\mu g/g^2$	Z-accelerometer
45	MRC	0.7	gm-cm	pendulosity parameter
46	QMIS(1)	1.0	unity	compensation misalignment transformation matrix from X, Y, Z accelerometer output axis to X, Y, Z body axes
47	QMIS(2)	0.0	unity	
48	QMIS(3)	0.0	unity	
49	QMIS(4)	0.0	unity	
50	QMIS(5)	1.0	unity	
51	QMIS(6)	0.0	unity	
52	QMIS(7)	0.0	unity	
53	QMIS(8)	0.0	unity	
54	QMIS(9)	1.0	unity	

Index	Variable	Default Value	Unit	Description
55	SFMO(1)	10.0	ppm	X-accelerometer compensation negative scale-factor error
56	SFMO(2)	10.0	ppm	Y-accelerometer compensation negative scale-factor error
57	SFMO(3)	10.0	ppm	Z-accelerometer compensation negative scale-factor error
58	SFP1(1)	0.0	ppm/g	X-accelerometer compensation positive rate sensitivity of scale-factor errors
59	SFP1(2)	0.0	ppm/g	Y-accelerometer compensation positive rate sensitivity of scale-factor errors
60	SFP1(3)	0.0	ppm/g	Z-accelerometer compensation positive rate sensitivity of scale-factor errors
61	SFM1(1)	0.0	ppm/g	X-accelerometer compensation negative rate sensitivity of scale-factor errors
62	SFM1(2)	0.0	ppm/g	Y-accelerometer compensation negative rate sensitivity of scale-factor errors
63	SFM1(3)	0.0	ppm/g	Z-accelerometer compensation negative rate sensitivity of scale-factor errors
64	RX(1)	1.01	ft	compensation position vector (vehicle cg to X-accelerometer origin) in body coordinates
65	RX(2)	0.0	ft	compensation position vector (vehicle cg to X-accelerometer origin) in body coordinates
66	RX(3)	0.0	ft	compensation position vector (vehicle cg to X-accelerometer origin) in body coordinates
67	RY(1)	0.89	ft	compensation position vector (vehicle cg to Y-accelerometer origin) in body coordinates
68	RY(2)	0.19	ft	compensation position vector (vehicle cg to Y-accelerometer origin) in body coordinates
69	RY(3)	0.0	ft	compensation position vector (vehicle cg to Y-accelerometer origin) in body coordinates
70	RZ(1)	0.91	ft	compensation position vector (vehicle cg to Z-accelerometer origin) in body coordinates
71	RZ(2)	0.0	ft	compensation position vector (vehicle cg to Z-accelerometer origin) in body coordinates
72	RZ(3)	0.19	ft	compensation position vector (vehicle cg to Z-accelerometer origin) in body coordinates
73	IXX	5.0	gm-cm^2	moment of inertial about output axis

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.0	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	---	not used

(4) Call-Line Data

$$(\overbrace{\mathrm{T,\ IENDF,\ D\bar{V}O,\ DT\overline{HE}TA}}^{\mathrm{Input}},\ \underbrace{\mathrm{D\bar{V}A}}_{\mathrm{Output}})$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
$D\bar{V}O$	ft/s	REAL	quantized integral of vehicle specific force along the X, Y, Z accelerometer input axes (over the fast cycle accumulated over slow cycle)
$DT\overline{HE}TA$	rad	REAL	quantized integral of angular rate about the X, Y, Z accelerometer input axes (over the fast cycle accumulated over slow cycle)

(b) Call-Line Output

Variable	Units	Data Type	Description
$\overline{DVA}$	ft/s	REAL	quantized integral of vehicle specific force along X, Y, Z accelerometer input axes (accumulated and compensated)

(5) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE

Define the matrix, QABO from the second rows (i.e., the output axes of the X, Y, and Z accelerometer) of QABX, QABY, and QABZ, respectively,

$$QABO = \begin{bmatrix} QABX(4) & QABX(5) & QABX(6) \\ QABY(4) & QABY(5) & QABY(6) \\ QABZ(4) & QABZ(5) & QABZ(6) \end{bmatrix}$$

Convert initialization data to computational units,

$$\overline{BIAS} = \overline{BIAS} * 1.E - 6*G$$

$$K\bar{I}I = KII * 1.E-6/G$$

$$IXX = IXX * 2.37E - 6$$

$$DELI = DELI * 2.37E - 6$$

$$MRC = MRC * 7.23E - 5$$

$$S\overline{P0} = SF\bar{P}0 * 1.E - 6$$

$$SP1 = SFP1 * 1.E - 6/G$$

$$S\bar{M}0 = SF\bar{M}0 * 1.E - 6$$

$$S\bar{M}1 = SF\bar{M}1 * 1.E - 6/G$$

where the conversion factors are

$$lb\text{-}ft^2 = 2.37E - 6 \ gm\text{-}cm^2$$

$$lb\text{-}ft = 7.23E - 5 \ gm\text{-}cm$$

$$unity = 1.E - 6 \ ppm$$

$$g \ ft/s^2 = (1.E - 6*G) \ \mu g$$

$$g \ ft/s^2 = (1.E - 6/G) \ g/g^2$$

The initialization switch is set, INITSW = 1 and the current simulation time is incremented

$$TCAC = T + DT$$

Now the subroutine returns to the main program.

(6) General Function

The following functions are performed every module operating cycle. The switch, K10, which was initialized to 0, is reset to 1 upon completion of the first pass after initialization.

The rate of change of angular velocity, $\overline{WDOT}$, resolved along the three accelerometer output axes, is bypassed on the first pass after initialization, but calculated on every subsequent pass

$$\overline{WDOT} = \overset{*}{QBA0} (\overline{DTHETA} - \overline{DTHETP})/DT$$

where

$\overline{DTHETP} = \overline{DTHETA}$ of the previous pass

The quantized incremental velocity (the quantized integral of the specific force), $\overline{DVO}$, for each of the X, Y, and Z accelerometers, in turn, is compensated for the effects of

- scale-factor error (positive and negative).
- bias.
- acceleration-squared sensitivity along the input axes.
- anisoinertia.
- lever-arm effect.
- output-axis couplings.

Scale-factor error is compensated by

$$\overline{DVO} = \overline{DVO}*(1 - \overline{SP0} + \overline{SP1}*\overline{DVO}/DT)$$

if DVO $\geq$ 0.0, for positive scale-factor error and by

$$\overline{DVO} = \overline{DVO}*(1 - \overline{SM0} + \overline{SM1}*\overline{DVO}/DT)$$

if $\overline{DVO}$ < 0.0, for negative scale-factor error.

Bias compensation is accomplished by use of the relation

$$\overline{DVO} = \overline{DVO} + \overline{BIAS} * DT$$

The components of $\overline{DVO}$ correspond to the X, Y, Z accelerometer input axes.

The cross products ($\overline{CROSS1}$ and $\overline{CROSS2}$) are calculated for the X-accelerometer to obtain the lever arm effect by

$$CROSS1 = \overline{DTHETA} \times \overline{RX}$$

$$CROSS2 = (\overline{DTHETA} \times \overline{CROSS1})/DT$$

and DTHETA is transformed from body to X-accelerometer coordinates every pass

$$\overline{DTHETZ} = QA\overset{*}{B}X * \overline{DTHETA}$$

where the components are

1. input axis
2. output axis
3. pendulous axis

while the caculation of the cross product, CROSS3(1), for the X accelerometer is bypassed only on the first pass after initialization and

$$CROSS3(1) = (DDTH(2)*RX(3) - DDTH(3)*RX(2))/DT$$

where

$$DDTH(2) = DTHETA(2) - DTHETP(2)$$

$$DDTH(3) = DTHETA(3) - DTHETP(3)$$

Similarly, for the Y-accelerometer

$$\overline{\text{CROSS1}} = \overline{\text{DTHETA}} \times \overline{\text{RY}}$$

$$\overline{\text{CROSS2}} = (\overline{\text{DTHETA}} \times \overline{\text{CROSS1}})/\text{DT}$$

and

$$\overline{\text{DTHETZ}} = \text{QABY}^{*} * \overline{\text{DTHETA}}$$

and

CROSS3(2) = (DDTH(3)*RY(1) - DDTH(1)*RY(3))/DT

where

DDTH(3) = DTHETA(3) - DTHETP(3)

DDTH(1) = DTHETA(1) - DTHETP(1)

Also, for the Z accelerometer

$$\overline{\text{CROSS1}} = \overline{\text{DTHETA}} \times \overline{\text{RZ}}$$

$$\overline{\text{CROSS2}} = (\overline{\text{DTHETA}} \times \overline{\text{CROSS1}})/\text{DT}$$

and

$$\overline{\text{DTHETZ}} = \text{QABZ}^{*} * \overline{\text{DTHETA}}$$

and

CROSS3(3) = (DDTH(1)*RZ(2) - DDTH(2)*RZ(1))/DT

where

DDTH(1) = DTHETA(1) - DTHETP(1)

DDTH(2) = DTHETA(2) - DTHETP(2)

The lever-arm effect is compensated by the terms $\overline{CROSS2}$ and $\overline{CROSS3}$ on the incremental velocity

$$\overline{DVO} = \overline{DVO} - \overline{CROSS3} - \overline{CROSS2}$$

The acceleration-squared sensitivity along the input axis is compensated by the term

$$\overline{KII} * \overline{DVO}^2/DT$$

for the X, Y, and Z accelerometers on every pass.

The anisoinertia is compensated by the term

$$DELI*DTHETZ(3)*DTHETZ(1)/(DT*MRC)$$

for the X, Y, and Z accelerometers using the pendulous (3) and input (1) components on every pass.

The acceleration-squared sensitivity along the input axis and the anisoinertia are added to the incremental velocity for

$$\overline{DVO} = \overline{DVO} + \overline{KII}*\overline{DVO}^2/DT + DELI*DTHETZ(3)*DTHETZ(1)/(DT*MRC)$$

along the X, Y, and Z accelerometer input axes.

Finally, the output-axis coupling is compensated on all passes, except the first pass after initialization for the X, Y, and Z accelerometers, and added to the incremental velocity

$$\overline{DVO} = \overline{DVO} + IXX * \overline{WDOT}/MRC$$

At the end of the first pass after initialization and compensation of the $\overline{DVO}$ along the X, Y, and Z accelerometer input axes, the switch K10 is reset, K10 = 1.

On every pass the quantized integral at angular rate is restored in $DTH\overline{E}TP$ for use in the next pass

$$DTH\overline{E}TP = DTH\overline{E}TA$$

Finally, the quantized integral of vehicle specific force (incremental velocity) along the X, Y, and Z accelerometer input axes, $\overline{DVA}$, is compensated for accelerometer misalignment

$$\overline{DVA} = QM\overset{*}{I}S * \overline{DVO}$$

The compensated incremental velocity, $\overline{DVA}$, is printed and the simulation time incremented

$$TCAC = T + DT$$

The subroutine now returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass, the title "ACC COMPENSATION INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for print control logic. The printed output is as follows

Variable	Units	Description
$\overline{DV}A$	ft/s	compensated, quantized integral of vehicle specific force (incremental velocity) along X, Y, and Z accelerometer input axes plus errors accumulated over the slow cycle

(7) Subroutines Called (See Section 2.3.15)

MXV = matrix by vector product

2.3.10 GYRO COMPENSATION MODULE (GCOMP)

(1) General Description

This module computes the compensation for gyro bias, scale-factor error, acceleration-sensitive terms, acceleration-squared sensitivities, scale-factor variation with rate, output-axis coupling, anisoinertia, and gyro misalignments.

(2) Gyro Compensation Module Flow Diagram

Figure 1 depicts the general logic flow of the gyro compensation module.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 69

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.02	s	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	---	not used
6	SPARE2	0.0	---	not used

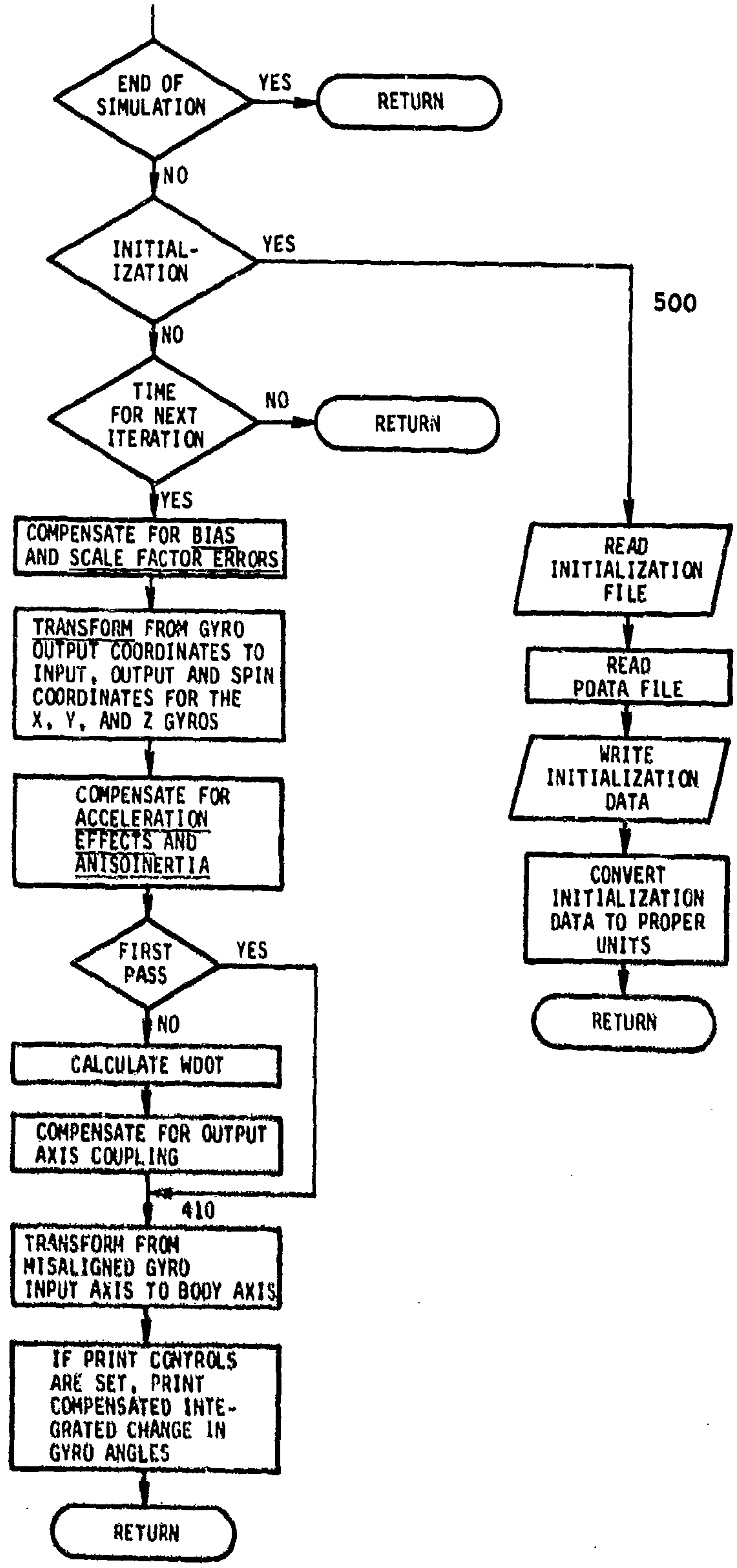

Figure 1. Gyro compensation module.

Index	Variable	Default Value	Units	Description
7	QMIS(1)	1.0	unity	compensating transformation from X, Y, Z gyro input axis to X, Y, Z body axes
8	QMIS(2)	0.0	unity	
9	QMIS(3)	0.0	unity	
10	QMIS(4)	0.0	unity	
11	QMIS(5)	1.0	unity	
12	QMIS(6)	0.0	unity	
13	QMIS(7)	0.0	unity	
14	QMIS(8)	0.0	unity	
15	QMIS(9)	1.0	unity	
16	-------	---	---	----
17	SFPO(1)	-30.0	ppm	compensation positive scale-factor error X gyro.
18	SFPO(2)	-30.0	ppm	compensation positive scale-factor error Y gyro.
19	SFPO(3)	-30.0	ppm	compensation positive scale-factor error Z gyro.
20	BIAS(1)	-0.34	deg/h	compensation bias error for X gyro.
21	BIAS(2)	-0.34	deg/h	compensation bias error for Y gyro.
22	BIAS(3)	-0.34	deg/h	compensation bias error for Z gyro.
23	SFMO(1)	-44.0	ppm	computation negative scale-factor error X gyro
24	SFMO(2)	-44.0	ppm	computation negative scale-factor error Y gyro
25	SFMO(3)	-44.0	ppm	computation negative scale-factor error Z gyro

Index	Variable	Default Value	Units	Description
26	QGBX(1)	1.0	unity	
27	QGBX(2)	0.0	unity	
28	QGBX(3)	0.0	unity	
29	QGBX(4)	0.0	unity	
30	QGBX(5)	0.0	unity	compensation transformation from X, Y, Z gyro output (body) axes to input, output, spin axes of X gyro
31	QGBX(6)	1.0	unity	
32	QGBX(7)	0.0	unity	
33	QGBX(8)	-1.0	unity	
34	QGBX(9)	0.0	unity	
35	QGBY(1)	0.0	unity	
36	QGBY(2)	1.0	unity	
37	QGBY(3)	0.0	unity	
38	QGBY(4)	1.0	unity	
39	QGBY(5)	0.0	unity	compensation transformation from X, Y, Z gyro output (body) axes to input, output, spin axes of Z gyro
40	QGBY(6)	0.0	unity	
41	QGBY(7)	0.0	unity	
42	QGBY(8)	0.0	unity	
43	QGBY(9)	-1.0	unity	
44	QGBZ(1)	0.0	unity	
45	QGBZ(2)	0.0	unity	compensation transformation from X, Y, Z gyro output (body) axes to input, output, spin axes of Z gyro
46	QGBZ(3)	1.0	unity	
47	QGBZ(4)	1.0	unity	

Index	Variable	Default Value	Units	Description		
48	QGBZ(5)	0.0	unity			
49	QGBZ(6)	0.0	unity	compensation transformation from X, Y, Z gyro output (body) axes to input, output, spin axes of Z gyro		
50	QGBZ(7)	0.0	unity			
51	QGBZ(8)	1.0	unity			
52	QGBZ(9)	0.0	unity			
53	KO(1)	-0.037	deg/h/g	compensation output axis acceleration sensitivity coefficient	X	
54	KO(2)	-0.037	deg/h/g		Y	gyro.
55	KO(3)	-0.037	deg/h/g		Z	
56	KI(1)	1.16	deg/h/g	compensation input-axis acceleration sensitive coefficient	X	
57	KI(2)	1.16	deg/h/g		Y	gyro.
58	KI(3)	1.16	deg/h/g		Y	
59	KS(1)	-0.66	deg/h/g	compensation spin-axis acceleration sensitive coefficient	X	
60	KS(2)	-0.66	deg/h/g		Y	gyro.
61	KS(3)	-0.66	deg/h/g		Z	
62	KII(1)	0.0	$deg/h/g^2$	compensation input-axis acceleration-squared sensitive coefficient	X	
63	KII(2)	0.0	$deg/h/g^2$		Y	gyro.
64	KII(3)	0.0	$deg/h/g^2$		Z	
65	H	151000.	$gm\text{-}cm^2/s$	compensation gyro angular momentum		
66	IXX	226.	$gm\text{-}cm^2$	compensation moment of inertia about output axis		
67	KSS(1)	0.0	$deg/h/g^2$	compensation spin-axis acceleration-squared sensitive coefficient	X	
68	KSS(2)	0.0	$deg/h/g^2$		Y	gyro.
69	KSS(3)	0.0	$deg/h/g^2$		Z	

Index	Variable	Default Value	Units	Description
70	KIO(1)	0.0	deg/h/g^2	compensation input - output axis acceleration-squared sensitive coefficient X gyro.
71	KIO(2)	0.0	deg/h/g^2	compensation input - output axis acceleration-squared sensitive coefficient Y gyro.
72	KIO(3)	0.0	deg/h/g^2	compensation input - output axis acceleration-squared sensitive coefficient Z gyro.
73	KOS(1)	-0.15	deg/h/g^2	compensation output - spin axis acceleration-squared sensitive coefficient X gyro.
74	KOS(2)	-0.15	deg/h/g^2	compensation output - spin axis acceleration-squared sensitive coefficient Y gyro.
75	KOS(3)	-0.15	deg/h/g^2	compensation output - spin axis acceleration-squared sensitive coefficient Z gyro.
76	KIS(1)	0.06	deg/h/g^2	compensation input - spin axis acceleration-squared sensitivie coefficient X gyro.
77	KIS(2)	0.06	deg/h/g^2	compensation input - spin axis acceleration-squared sensitivie coefficient Y gyro.
78	KIS(3)	0.06	deg/h/g^2	compensation input - spin axis acceleration-squared sensitivie coefficient Z gyro.
79	DELI	14.0	gm-cm^2	gyro spin-axis minus input-axis moment of inertia
80	SFM1(1)	0.0	ppm/deg/s	compensation negative scale-factor error varying with rate X gyro.
81	SFM1(2)	0.0	ppm/rad/s	compensation negative scale-factor error varying with rate Y gyro.
82	SFM1(3)	0.0	ppm/rad/s	compensation negative scale-factor error varying with rate Z gyro.
83	SFP1(1)	0.0	ppm/rad/s	compensation positive scale-factor error varying with rate X gyro.
84	SFP1(2)	0.0	ppm/rad/s	compensation positive scale-factor error varying with rate Y gyro.
85	SFP1(3)	0.0	ppm/rad/s	compensation positive scale-factor error varying with rate Z gyro.
86	MODPDT	6.0	s	module print interval

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.729211514E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	---	not used

(4) Call-Line Data

$$\overbrace{\text{(T, IENDF, DTHETO, DVA,}}^{\text{Input}}$$

$$\underbrace{\text{DTHETZ)}}_{\text{Output}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
DTHETO	rad	REAL	quantized integral of angular rate about X, Y, Z gyro input axes (fast-cycle computation, accumulated over slow cycle)
DVA	ft/s	REAL	compensated quantized incremental velocity (fast-cycle computations accumulated over slow cycle)

(b) Call-Line Output

Variable	Units	Data Type	Description
DTHETZ	rad	REAL	compensated, quantized integral of angular rate about X, Y, Z gyro input axes (fast-cycle computations accumulated over slow cycle)

(5) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE.
- Convert initialization data to computational units,

$$
\begin{array}{lcl}
S\overline{M}1 & = & SF\overline{M}1 * 1.E - 6 \\
S\overline{P}1 & = & SF\overline{P}1 * 1.E - 6 \\
S\overline{P}0 & = & SF\overline{P}0 * 1.E - 6 \\
S\overline{N}0 & = & SF\overline{N}0 * 1.E - 6 \\
BI\overline{A}S & = & BI\overline{A}S * 4.85E - 6 \\
\overline{KI} & = & \overline{KI} * 4.85E - 6/G \\
\overline{KO} & = & \overline{KO} * 4.85E - 6/G \\
\overline{KS} & = & \overline{KS} * 4.85E - 6/G \\
K\overline{II} & = & K\overline{II} * 4.85E - 6/G^2 \\
K\overline{I}S & = & K\overline{I}S * 4.85E - 6/G^2 \\
K\overline{O}S & = & K\overline{O}S * 4.85E - 6/G^2 \\
K\overline{S}S & = & K\overline{S}S * 4.85E - 6/G^2 \\
K\overline{I}O & = & K\overline{I}O * 4.85E - 6/G^2 \\
IXX & = & IXX * 2.37E - 6 \\
DELI & = & DELI * 2.37E - 6 \\
H & = & H * 2.37E - 6
\end{array}
$$

where the conversion factors are

$$\mathrm{lb\text{-}ft}^2 = 2.37\mathrm{E} - 6\ \mathrm{gm\text{-}cm}^2$$

$$\mathrm{unity} = 1.\mathrm{E} - 6\ \mathrm{ppm}$$

$$\mathrm{deg/h/g} \quad 4.85\mathrm{E} - 6\ \mathrm{rad/s/g}$$

$$\mathrm{deg/h/g}^2 = 4.85\mathrm{E} - 6\ \mathrm{rad/s/g}^2$$

Define the matrix, $\mathrm{QO\overset{*}{B}G}$, from the second row (i.e., the output axes of the X, Y, and Z gyros) of $\mathrm{QG\overset{*}{B}X}$, $\mathrm{QG\overset{*}{B}Y}$, and $\mathrm{QG\overset{*}{B}Z}$, respectively,

$$\mathrm{QO\overset{*}{B}G} = \begin{vmatrix} \mathrm{QGBX(4)} & \mathrm{QGBX(5)} & \mathrm{QGBX(6)} \\ \mathrm{QGBY(4)} & \mathrm{QGBY(5)} & \mathrm{QGBY(6)} \\ \mathrm{QGBZ(4)} & \mathrm{QGBZ(5)} & \mathrm{QGBZ(6)} \end{vmatrix}$$

The initialization switch is set, INITSW = 1, and the current simulation time is incremented

$$\mathrm{TCGY} = \mathrm{T} + \mathrm{DT}$$

Now the subroutine returns to the main program.

(b) <u>General Function</u>

The following functions are performed every module operating cycle.

The quantized integral of the angular rate about the X, Y, and Z gyro input axes are compensated for positive scale-factor error when $\overline{\text{DTHETO}} \geq 0.0$ by the factor

$$\text{T1} = 1. + \overline{\text{SP0}} + \overline{\text{SP1}} * \overline{\text{DTHETO}}/\text{DT}$$

and for negative scale-factor error when DTHETO < 0.0 by the factor

$$\text{T1} = 1. + \overline{\text{SM0}} + \overline{\text{SM1}} * \overline{\text{DTHETO}}/\text{DT}$$

A bias correction is added,

$$\overline{\text{DTHETO}} = \overline{\text{DTHETO}} * \text{T1} + \overline{\text{BIAS}}$$

The quantized integral of angular rate, $\overline{\text{DTHETO}}$, is compensated for acceleration effects for X, Y, and Z gyro, in turn.

First, for the X gyro, the vector DVA is transformed from the X, Y, Z accelerometer input-axis coordinates to accelerometer (input, output, and pendulous) coordinates

$$\overline{\text{DVG}} = \text{QGBX}^{*} * \overline{\text{DVA}}$$

Also, for the X gyro, the vector $\overline{\text{DTHETO}}$ is similarly transformed from the X, Y, Z gyro input-axis coordinates to gyro (input, output, spin) coordinates

$$\overline{\text{DTHETG}} = \text{QGBX}^{*} * \overline{\text{DTHETO}}$$

Similarly, for the Y gyro

$$\overline{\text{DVG}} = \text{QGBY}^{*} * \overline{\text{DVA}}$$

and

$$\overline{DTHETG} = QGBY^* * \overline{DTHETO}$$

and for the Z gyro

$$\overline{DVG} = QGBZ^* * \overline{DVA}$$

$$\overline{DTHETG} = QGBZ^* * \overline{DTHETO}$$

From the transformed vectors, $\overline{DVG}$ and $\overline{DTHETG}$, the acceleration effects and anisoinertia compensation are computed for the X, Y, and Z gyros

$$\begin{aligned} \overline{DTHETO} = \ & \overline{DTHETO} + \overline{KI}*DVG(1) + \overline{KO}*DVG(2) + \overline{KS}*DVG(3) \\ & +(\overline{KOS}*DVG(2)*DVG(3) + \overline{KSS}*DVG(3)^2 + \overline{KII}*DVG(1)^2 \\ & + \overline{KIS}*DVG(1)*DVG(3) + \overline{KIO}*DVG(1) * DVG(2))/DT \\ & - DELI*DTHETG(1) * DTHETG(3)/(H*DT) \end{aligned}$$

Finally, output axis coupling compensation is accomplished by the following computation on all passes except the first pass after initialization, when K1 = 0. First, the difference between $\overline{DTHETO}$ and $\overline{DTH}$, where $\overline{DTH}$ = $\overline{DTHETO}$ of the previous pass is defined as

$$\overline{WDOT} = \overline{DTHETO} - \overline{DTH}$$

and $\overline{WDOT}$ is resolved about the X, Y, Z gyro output axes, as $\overline{DWO}$

$$\overline{DWO} = QOBG^* * \overline{WDOT}$$

and

$$\overline{DTHETO} = \overline{DTHETO} + IXX * \overline{DWO}/(H * DT)$$

with components about the X, Y, and Z gyro input axis.

On every pass, DTHETO is restored as DTH

$$\overline{DTH} = \overline{DTHETO}$$

for use in the subsequent pass. K10 set to 1 on the first pass after initialization.

The quantized integral of angular rate, DTHETZ, about the X, Y, and Z gyro input axes is compensated for misalignment

$$\overline{DTHETZ} = \overset{*}{QMIS} * \overline{DTHETO}$$

The vector, $\overline{DTHETZ}$, is printed and the simulation time incremented

$$TCGY = T + DT$$

The subroutine now returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass, the title "GYR COMPENSATION INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for printe control logic. The printed output is, as follows

Variable	Units	Description
$\overline{DTHETZ}$	rad	compensated quantized integral of angular rate about the X, Y, Z gyro input axes over the slow computation cycle plus errors

(7) <u>Subroutines Called</u> (See Section 2.3.15)

MXV = matrix by vector product

2.3.11 LASER GYRO COMPENSATION MODULE (GCOMP)

(1) General Description

The compensation module accepts incremental rotations from the simulated laser gyros and produces compensated incremental rotations in body coordinates. The module contains compensation for the following error sources:

- gyro input axis misalignment
- scale factor error
- scale factor turn-on transient
- gyro bias
- turn-on transient drift

(2) Gyro Module Flow Diagram

Figure 1 is a flow diagram of the laser gyro compensation module. The module is initialized on the first pass through the routine. The compensation module initialization file (IFILE) is read and the compensation parameters are converted to internal program units. The program then performs the normal module operations. An analytical description of the laser compensation equations is given in Volume II, Section 5.2.3.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number : 69

FORTRAN format: I5, F20.10

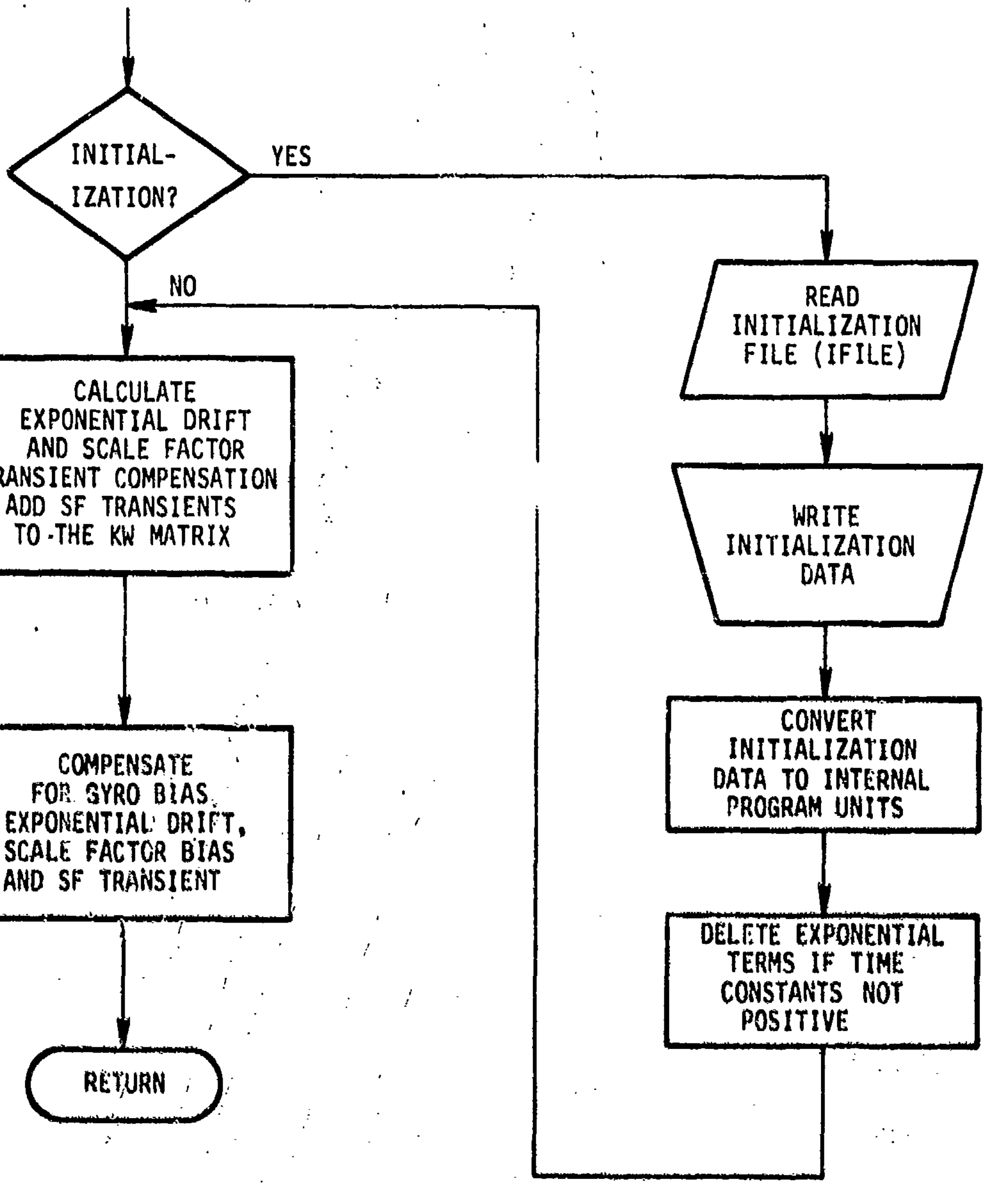

Figure 1. Laser Gyro Compensation Module Flow Diagram

Index	Variable	Default Value	Units	Description
1	DT	0.02	s	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE1	0.0	---	not used
6	SPARE2	0.0	---	not used
7	MODPDT	6.0	s	module print interval
8	DB(1)	0.7	deg/hr	gyro bias
9	DB(2)	0.4	deg/hr	
10	DB(3)	-0.6	deg/hr	
11	DTA(1)	0.06	deg/hr	transient drift magnitude
12	DTA(2)	-0.03	deg/hr	
13	DTA(3)	-0.05	deg/hr	
14	DTC(1)	45.0	min	transient drift time constant
15	DTC(2)	45.0	min	
16	DTC(3)	45.0	min	
17	SFTA(1)	0.8	ppm	transient scale factor amplitude
18	SFTA(2)	-0.7	ppm	
19	SFTA(3)	1.0	ppm	
20	SFTC(1)	45.0	min	transient scale factor time constant
21	SFTC(2)	45.0	min	
22	SFTC(3)	45.0	min	
23	KW(1)	1.0	ppm	gyro SF and gyro input axis (IA) misalignment matrix • main diagonal = SF bias (ppm) • off diagonal = IA mis (microrad ppm)
24	KW(2)	20.0	ppm	
25	KW(3)	50.0	ppm	
26	KW(4)	40.0	ppm	
27	KW(5)	1.5	ppm	
28	KW(6)	30.0	ppm	
29	KW(7)	-30.0	ppm	
30	KW(8)	-40.0	ppm	
31	KW(9)	-1.0	ppm	

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: 15, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.729211514E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	PBUF(1)	0.0	---	not used

(4) Call-Line Data

$$\overbrace{(T,\ IENDF,\ \overline{DTHETO},\ \overline{DVA},}^{\text{Input}}\ \underbrace{\overline{DTHETZ})}_{\text{Output}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	Current simulation time
IENDF	logical	INTEGER	last-pass indicator
$\overline{\text{DTHETO}}$	rad	REAL	quantized integral of angular rate about X, Y, Z gyro input axes (fast-cycle computation, accumulated over slow cycle)
$\overline{\text{DVA}}$	---	REAL	not used

(b) Call-Line Output

Variable	Units	Data Type	Description
$D\overline{THE}TZ$	rad	REAL	compensated, quantized integral of angular rate about X, Y, Z gyro input axes (fast-cycle computations accumulated over slow cycle)

(5) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE.

- Convert the initialization data to internal program units. Internal units are: radians, feet, and seconds.

$\overline{DB} = \overline{DB} * DTR/3600.$

$\overline{DTA} = \overline{DTA} * DTR/3600.$

$\overline{DTC} = \overline{DTC} * 60.$

$\overline{SFTA} = \overline{SFTA} * 1.E-6.$

$\overline{SFTC} = \overline{SFTC} * 60.$

$\overset{*}{KW} = \overset{*}{KW} * 1.E-6$

where DTR equals degrees-to-radians conversion factor.

- Compute the exponential drift and exponential scale factor coefficients.

$\overline{EXPD} = EXP(-DT/\overline{DTC})$

$\overline{EXPSF} = EXP(-DT/\overline{SFTC})$

The vector DTC contains the time constants of the exponential drifts and SFTC contains the time constants of the exponential scale factors. The elements of $\overline{EXPD}$ and $\overline{EXPSF}$ are set to zero if the corresponding time constants are nonpositive. The difference equation for each of the terms is of the form

$$D_T(n+1) = D_T(n) * \overline{EXPD}$$

$$S_T(N+1) = S_T(n) * \overline{EXPSF}$$

Thus, the individual transients are set to zero if the associated time constant is not positive. This allows the user to delete compensation for any of the transient terms by setting the amplitude and time constant to zero.

The initialization switch is set, INITSW = 1, and the current simulation time is incremented

$$TCGY = T + DT$$

Now the subroutine returns to the main program.

(b) General Function

The following functions are performed every module operating cycle.

- Compute the transient drift and scale factor compensation.

$$\overline{DTA} = \overline{DTA} * \overline{EXPD}$$

$$\overline{SFTA} = \overline{SFTA} * \overline{EXPSF}$$

The transient terms are set to zero if the magnitude becomes less than 1.E-10 to prevent underflow.

- Add the scale factor transient to the scale factor bias. Insert the sum on the main diagonal of the rate sensitive matrix $\overset{*}{KW}$.

$$KW(K) = SF(I) + SFTA(I), \text{ where } K = \text{diagonal elements}$$

The off diagonal elements of $\overset{*}{KW}$ contain the IA misalignments.

- Compute the compensated incremental rotation $\overline{DTHETZ}$.

$$\overline{DTHETZ} = \left[(\overset{*}{I} - \overset{*}{KW}) \; \overline{DTHETO} - (\overline{DB} + \overline{DTA}) \; DT \right]$$

where

$\overline{DTHETO}$ = quantized angles from the simulated laser gyros
$\overline{DB}$ = gyro bias compensation
$\overline{DTA}$ = transient drift compensation
$\overset{*}{KW}$ = angular rate sensitive coefficient compensation
$\overset{*}{I}$ = identity matrix

The vector, $\overline{DTHETZ}$, is printed and the simulation time incremented

$$TCGY = T + DT$$

The subroutine now returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass, the title "LASER GYRO COMPENSATION INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for printe control logic. The printed output is, as follows

Variable	Units	Description
$\overline{DTHETZ}$	rad	Compensated quantized integral of angular rate about the X, Y, Z gyro input axes over the slow computation cycle plus errors

(7) Subroutines Called (See Section 2.3.15)

MXV = matrix by vector product

2.3.12 ATTITUDE AND VELOCITY ALGORITHM MODULE (ALG)

(1) General Description

The first segment of this module, the "attitude" algorithm processes the incremental angle data, $\underline{\Delta\theta}$, (from the gyro compensation module) to update the body-to-inertial transformation, C_b^i, the initial value of which is computed in the initilization pass. Either a quaternion or a direction cosine matrix update, of first through third order, may be employed, as specified by the user.

The second segment uses C_b^i to transform the incremental velocities, $\underline{\Delta V}$, (from the accelerometer compensation module) to the inertial frame, where they may be summed, and passed on to the navigation module.

An additional transformation from the inertial frame to the navigator's earth-fixed frame, $C_i^{e'}$, is applied to C_b^i before it is passed on to the latter module for use in attitude computation.

(2) Attitude and Velocity Algorithm Module Flow Diagram

The general logic flow of the ALG module is shown in Figure 1.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 70

FORTRAN format: I5, F20.10

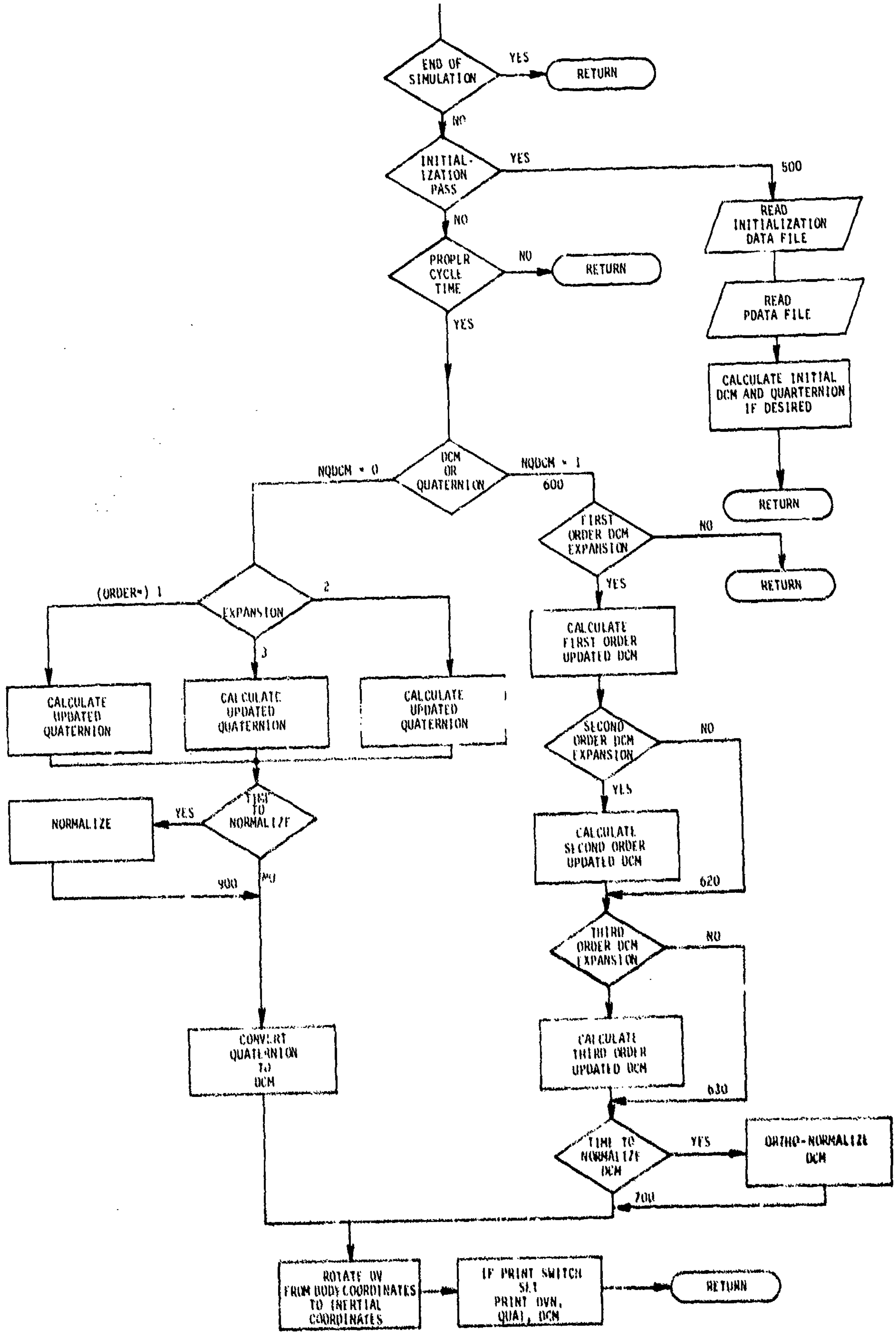

Figure 1. Velocity and attitude algorithm module flow diagram

Index	Variable	Default Value	Units	Description
1	DT	0.02	secs	module operating frequency
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	---	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	QORDCM	1.0	logical	algorithm switch {0 - quaternion; 1 - direction cosine matrix}
6	MODPDT	6.0	secs	module print interval
7-10	---	---	---	not used
11	ORDER	3.0	logical	order specification for quaternion or direction cosine (dcm) update {first-order=1; second-order=2; third-order=3}
12	DTNRM	4.0	secs	time increment between quaternion or dcm normalization calculation
13	DTSLOW	0.0	---	not used
14	LATERR	0.0	deg	initial vehicle latitude error
15	LONERR	0.0	deg	initial vehicle longitude error
16	WANERR	0.0	deg	initial vehicle wander angle error
17	PITERR	0.0	deg	initial vehicle pitch error
18	ROLERR	0.0	deg	initial vehicle roll error
19	YAWERR	0.0	deg	initial vehicle yaw error

(b) <u>Common Initialization File (PFILE)</u>

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/sec	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/sec^2	nominal gravity
4	PRNTDT	6.0	secs	printing frequency
5	ILAT	0.0	deg	initial latitude
6	ILON	0.0	deg	initial longitude

Index	Variable	Default Value	Units	Description
7	WANDER	0.0	deg	initial wander angle
8	---	0.0	---	not used
9	ROLL	0.0	deg	initial roll angle
10	PITCH	0.0	deg	initial pitch angle
11	YAW	0.0	deg	initial yaw angle

(4) Call Line Data

$$\overbrace{\text{T, IENDF, DTHETI, DV,}}^{\text{INPUT}}$$

$$\underbrace{\overbrace{\text{DVN}}^{\text{IN}}\text{, DCM}}_{\text{OUTPUT}}$$

(a) Call Line Input

Variable	Units	Data Type	Description
T	secs	REAL	current simulation time
IENDF	logical	INTEGER	last pass indicator
DTHETI	rad	REAL	compensated quantized integral of angular rate about X,Y,Z gyro input axes (fast cycle computations accumulated over slow cycle)
DV	ft/sec	REAL	compensated quantized integral of vehicle specific force along X,Y,Z accelerometer input axes (fast cycle computations accumulated over slow cycle)

(b) Call Line Output

Variable	Units	Data Type	Description
$\overline{DVN}$	ft/sec	REAL	quantized integral of vehicle specific force in the inertial frame of the local level navigation module (LLN)
$\overset{*}{DCM}$	units	REAL	direction cosine matrix (body to earth-fixed frame transformation)

(5) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE
  NQDCM = QORDCM
  NORDER = ORDER
- Initialize
  $\overline{DVN}$ = 0.0
  $\underset{\sim}{DTHPRE}$ = 0.0
- Define the following quantities,

  TNORM = T + DTNRM

  and convert the angles in degrees to radians,

  COSP = COS ((PITCH+PITERR)/RDTODG)
  COSY = COS ((YAW+YAWERR)/RDTODG)
  COSR = COS ((ROLL+ROLLERR)/RDTODG)
  COSLAT = COS ((ILAT+LATERR)/RDTODG)
  COSLON = COS ((ILON+LONERR)/RDTODG)
  COSW = COS ((WANDER+WANERR)/RDTODG)

$$
\begin{aligned}
\text{SINP} &= \text{SIN ((PITCH+PITERR)/RDTODG)}\\
\text{SINY} &= \text{SIN ((YAW+YAWERR)/RDTODG)}\\
\text{SINR} &= \text{SIN ((ROLL+ROLLERR)/RDTODG)}\\
\text{SINLAT} &= \text{SIN ((ILAT+LATERR)/RDTODG)}\\
\text{SINLON} &= \text{SIN ((ILON+LONERR)/RDTODG)}\\
\text{SINW} &= \text{SIN ((WANDER+WANERR)/RDTODG)}
\end{aligned}
$$

where the conversion factor,

$$\text{RDTODG} = 57.29577951 \text{ deg/rad}$$

Further the matrix, $\overset{*}{QPB}$, a body to computational frame transformation matrix is defined,

$$
\overset{*}{QPB} = \left|\begin{array}{c|c|c}
\text{COSP*SINY} & \text{-SINR*SINP*SINY-COSY*COSR} & \text{-COSR*SINP*SINY+COSY*SINR}\\
\text{COSP*COSY} & \text{-SINR*SINP*COSY+COSR*SINY} & \text{-COSR*SINP*COSY-SINR*SINY}\\
\text{SINP} & \text{COSP*SINR} & \text{COSP*COSR}
\end{array}\right|
$$

from the single axis rotations (Appendix A of Volume II)

$$\overset{*}{QPB} = Z(-\pi/2)\ X(-\pi)\ Z(-Y)\ Y(-P)\ X(-R)\ X(-\pi)$$

The matrix, $\overset{*}{QIB}$, a computational to inertial frame transformation matrix is defined,

$$
\overset{*}{QIP} = \left|\begin{array}{c|c|c}
\text{COSLON*COSW-SINLON*SINW*SINLAT} & \text{-SINW*COSLON-COSW*SINLON*SINLAT} & \text{COSLAT*SINLON}\\
\text{SINW*COSLAT} & \text{COSW*COSLAT} & \text{SINLAT}\\
\text{-COSW*SINLON-SINW*SINLAT*COSLON} & \text{SINW*SINLON-SINLAT*COSLON*COSW} & \text{COSLAT*COSLON}
\end{array}\right|
$$

The body to inertial frame transformation matrix, $\overset{*}{QIB}$, is defined by the matrix multiplication,

$$\overset{*}{QIB} = \overset{*}{QIP} * \overset{*}{QPB}$$

The vector, $\overline{QUAT}$, is initialized as,

$$
\begin{aligned}
\text{QUAT(1)} &= ((1+\text{QIB(1)}+\text{QIB(5)} + \text{QIB(9)})/4)^{1/2}\\
\text{QUAT(2)} &= (\text{QIB(8)}-\text{QIB(6)})/(4*\text{QUAT(1)})\\
\text{QUAT(3)} &= (\text{QIB(3)}-\text{QIB(7)})/(4*\text{QUAT(1)})\\
\text{QUAT(4)} &= (\text{QIB(4)}-\text{QIB(2)})/(4*\text{QUAT(1)})
\end{aligned}
$$

for a quaternion update or when NQDCM = 1, for a direction cosine matrix update, $\overline{QUAT}$ is initialized as

$$\overline{QUAT} = (0.0,\ 0.0,\ 0.0,\ 0.0)$$

The vector, $\overline{DTHP}$, the previous pass value of $\overline{DTHETI}$ is

$$\overline{DTHP} = (0.0,\ 0.0,\ 0.0)$$

The initialization switch is reset

$$INITSW = 1$$

and the simulation time incremented

$$TALG = T + DT$$

The subroutine now returns to the main program.

(b) <u>General Function</u>

The following functions are performed every module operating cycle.

Define $\overset{*}{DCMOLD}$, the direction cosine matrix of the previous pass, as

$$\overset{*}{DCMOLD} = \overset{*}{QIB}$$

for use in the interpolated values of $\overset{*}{DCM}$ and $\overset{*}{DCMMID}$

The vectors $\overline{\text{DTHETA}}$ and $\overline{\text{DTHP}}$ are defined

$$\overline{\text{DTHETA}} = \overline{\text{DTHETI}}$$
$$\overline{\text{DTHP}} = \overline{\text{DTHETI}}$$

A direction cosine matrix, $\overset{*}{\text{DCM}}$, is computed by one of two update methods: when NQDCM = 0, a quaternion update calculation or NQDCM = 1, a direction cosine matrix update calculation. In either case there is a choice of a first order, second order, or third order update.

The quaternion calculation proceeds with the definition of the matrix, $\overset{*}{\text{UDELT}}$

$$\overset{*}{\text{UDELT}} = \left|\begin{array}{c|c|c|c} \text{---} & -\text{DTHETA}(1) & -\text{DTHETA}(2) & -\text{DTHETA}(3) \\ \text{DTHETA}(1) & \text{---} & \text{DTHETA}(3) & -\text{DTHETA}(2) \\ \text{DTHETA}(2) & -\text{DTHETA}(3) & \text{---} & \text{DTHETA}(1) \\ \text{DTHETA}(3) & \text{DTHETA}(2) & -\text{DTHETA}(1) & \text{---} \end{array}\right|$$

where the diagonal elements are undefined. (See Vol. II, Section 7 for quaternion update discussion.)

For the first order quaternion update, when NORDER = 1, the terms of the quaternion, $\overline{\text{QUAT}}$, are

```
QUAT(1)  =  QUAT(1) + (-DTHETA(1)*QUAT(2)
                       -DTHETA(2)*QUAT(3)
                       -DTHETA(3)*QUAT(4))/2

QUAT(2)  =  QUAT(2) + ( DTHETA(1)*QUAT(1)
                       +DTHETA(3)*QUAT(3)
                       -DTHETA(2)*QUAT(4)/2

QUAT(3)  =  QUAT(3) + ( DTHETA(2)*QUAT(1)
                       -DTHETA(3)*QUAT(2)
                       +DTHETA(1)*QUAT(4))/2

QUAT(4)  =  QUAT(4) + ( DTHETA(3)*QUAT(1)
                       +DTHETA(2)*QUAT(2)
                       -DTHETA(1)*QUAT(3))/2
```

where the terms on the right refer to the values of $\overline{\text{QUAT}}$ on the previous pass.

The dot product, UDTH2, is defined for use in both the second and third order quaternion update as

$$\text{UDTHS} = -(\text{DTHETA}(1)^2 + \text{DTHETA}(2)^2 + \text{DTHETA}(3)^2)$$

The second order quaternion update is calculated, when NORDER = 2. The terms of the quaternion are expanded for clarity.

```
QUAT(1) = QUAT(1)*(1+UDTH2/8) + (-DTHETA(1)*QUAT(2)
                                 -DTHETA(2)*QUAT(3)
                                 -DTHETA(3)*QUAT(4))/2

QUAT(2) = QUAT(2)*(1+UDTH2/8) + ( DTHETA(1)*QUAT(1)
                                 +DTHETA(3)*QUAT(3)
                                 -DTHETA(2)*QUAT(4))/2

QUAT(3) = QUAT(3)*(1+UDTH2/8) + ( DTHETA(2)*QUAT(1)
                                 -DTHETA(3)*QUAT(2)
                                 +DTHETA(1)*QUAT(4))/2
```

```
QUAT(4) = QUAT(4)*(1+UDTH2/8) + ( DTHETA(3)*QUAT(1)
                                  +DTHETA(2)*QUAT(2)
                                  -DTHETA(1)*QUAT(3))/2
```

The third order quaternion update is calculated, when NORDER = 3. The terms of the quaternion are expanded for clarity. The cross product, $\overline{\text{LDTH}}$, is defined for the angles, $\overline{\text{DTHETA}}$, and $\overline{\text{DTHPRE}}$ of the previous pass $\overline{\text{DTHETA}}$.

$$\overline{\text{LDTH}} = \overline{\text{DTHPRE}} \text{ X } \overline{\text{DTHETA}}$$

These cross products are redefined in a matrix, $\text{LDEL}\overset{*}{\text{T}}\text{2}$.

$$\text{LDEL}\overset{*}{\text{T}}\text{2} = \begin{vmatrix} \text{---} & \text{-LDTH(1)} & \text{-LDTH(2)} & \text{-LDTH(3)} \\ \text{LDTH(1)} & \text{---} & \text{LDTH(3)} & \text{-LDTH(2)} \\ \text{LDTH(2)} & \text{-LDTH(3)} & \text{---} & \text{LDTH(1)} \\ \text{LDTH(3)} & \text{LDTH(2)} & \text{-LDTH(1)} & \text{---} \end{vmatrix}$$

where the diagonal elements are undefined.

```
QUAT(1) = QUAT(1) * (1+UDTH2/8)
         +QUAT(2) * (-(1/2+UDTH2/48)*DTHETA(1)-LDTH(1)/24)
         +QUAT(3) * (-(1/2+UDTH2/48)*DTHETA(2)-LDTH(2)/24)
         +QUAT(4) * (-(1/2+UDTH2/48)*DTHETA(3)-LDTH(3)/24)

QUAT(2) = QUAT(2) * (1+UDTH2/8)
         +QUAT(1) * ( (1/2+UDTH2/48)*DTHETA(1)+LDTH(1)/24)
         +QUAT(3) * ( (1/2+UDTH2/48)*DTHETA(3)+LDTH(3)/24)
         +QUAT(4) * (-(1/2+UDTH2/48)*DTHETA(2)-LDTH(2)/24

QUAT(3) = QUAT(3) * (1+UDTH2/8)
         +QUAT(1) * ( (1/2+UDTH2/48)*DTHETA(2)+LDTH(2)/24)
         +QUAT(2) * (-(1/2+UDTH2/48)*DTHETA(3)-LDTH(3)/24)
         +QUAT(4) * ( (1/2+UDTH2/48)*DTHETA(1)+LDTH(1)/24)

QUAT(4) = QUAT(4) * (1+UDTH2/8)
         +QUAT(1) * ( (1/2+UDTH2/48)*DTHETA(3)+LDTH(3)/24)
         +QUAT(2) * ( (1/2+UDTH2/48)*DTHETA(2)+LDTH(2)/24)
         +QUAT(3) * (-(1/2+UDTH2/48)*DTHETA(1)-LDTH(1)/24)
```

If NORDER $\neq$ 1, 2, or 3, the statement, "ORDER NOT PROPERLY SPECIFIED" is printed, the last pass switch is set, IENDF = 1, and the subroutine returns to the main program.

The simulation time is tested and if the interval DTNRM has elapsed or $T \geq TNORM$ the quaternion is normalized

$$\overline{QUAT} = \overline{QUAT}/\left[QUAT(1)^2 + QUAT(2)^2 + QUAT(3)^2\right]^{1/2}$$

and the normalization time is incremented

$$TNORM = T + DTNRM$$

The quaternion is converted to the direction cosine matrix, $\overset{*}{DCM}$.

First define:

$$
\begin{aligned}
Q1S &= QUAT(2)^2 \\
Q2S &= QUAT(3)^2 \\
Q3S &= QUAT(4)^2 \\
Q01 &= QUAT(1) * QUAT(2) \\
Q02 &= QUAT(1) * QUAT(3) \\
Q03 &= QUAT(1) * QUAT(4) \\
Q12 &= QUAT(2) * QUAT(3) \\
Q23 &= QUAT(3) * QUAT(4) \\
Q31 &= QUAT(4) * QUAT(2)
\end{aligned}
$$

Then

$$\overset{*}{DCM} = \left|\begin{array}{c|c|c} 1-2*(Q2S+Q3S) & 2*(Q12-Q03) & 2*(Q31+Q02) \\ 2*(Q12+Q03) & 1-2*(Q3S-Q1S) & 2*(Q23+Q01) \\ 2*(Q31-Q02) & 2*(Q23+Q01) & 1-2*(Q1S+Q2S) \end{array}\right|$$

An alternative to the quaternion update calculation is the direction cosine matrix update.

For the first order update, when NORDER = 1, the matrix, $\overset{*}{DTM}$ is defined.

$$\overset{*}{DTM} = \begin{vmatrix} 1 & -DTHETA(3) & DTHETA(2) \\ DTHETA(3) & 1 & -DTHETA(1) \\ -DTHETA(2) & DTHETA(1) & 1 \end{vmatrix}$$

For the second order direction cosine matrix update, when NORDER = 2, the following terms are calculated and added to the first order matrix, $\overset{*}{DTM}$.

$$\overset{*}{DTM} = \overbrace{\begin{vmatrix} DTM(1) & DTM(2) & DTM(3) \\ DTM(4) & DTM(5) & DTM(6) \\ DTM(7) & DTM(8) & DTM(9) \end{vmatrix}}^{\text{First Order}}$$

$$+1/2 \cdot \begin{vmatrix} -(DTHETA(2)^2+DTHETA(3)^2 & DTHETA(1)*DTHETA(2) & DTHETA(3)*DTHETA(1) \\ DTHETA(1)*DTHETA(2) & -(DTHETA(3)^2+DTHETA(1)^2) & DTHETA(2)*DTHETA(3) \\ DTHETA(3)*DTHETA(1) & DTHETA(2)+DTHETA(3) & -(DTHETA(1)^2+DTHETA(2)^2) \end{vmatrix}$$

For the third order direction cosine matrix update, when NORDER = 3, the following terms are caculated and added to the second order matrix, $\overset{*}{DTM}$,

First define the term, D

$$D = 0.16666666666666 * (DTHETA(1)^2+DTHETA(2)^2+DTHETA(3)^2)$$

Then define the third order terms C1, C2, and C3. In this version these terms are set to zero,

$$C1 = 0.0$$

$$C2 = 0.0$$

$$C3 = 0.0$$

due to the infinite accelerations generated in the available version of PROFGEN. The correct terms are defined in comment statements for the third order

$$C1 = 0.8333333333333*D-1*(DTHETA(3)*DTHPRE(2)-DTHETA(2)*DTHPRE(3))$$

$$C2 = 0.8333333333333*D-1*(-DTHETA(3)*DTHPRE(1)+DTHETA(1)*DTHPRE(3))$$

$$C3 = 0.8333333333333*D-1*(DTHETA(2)*DTHPRE(1)-DTHETA(1)*DTHPRE(2))$$

Thus,

$$\overset{*}{DTM} = \overbrace{\begin{vmatrix} DTM(1) & DTM(2) & DTM(3) \\ DTM(4) & DTM(5) & DTM(6) \\ DTM(7) & DTM(8) & DTM(9) \end{vmatrix}}^{\text{Second Order}}$$

$$+ \begin{vmatrix} 0 & D*DTHETA(3)-C3 & -D*DTHETA(2)+C2 \\ -D*DTHETA(3)+C3 & 0 & D*DTHETA(1)-C1 \\ D*DTHETA(2)-C2 & -D*DTHETA(1)+C1 & 0 \end{vmatrix}$$

If NORDER $\neq$ 1, 2, or 3, the statement, "ORDER NOT PROPERLY SPECIFIED" is printed, the last pass switch is set, IENDF = 1, and the subroutine returns to the main program.

The matrix, $\overset{*}{DTM}$, is multiplied by the body to inertial frame transformation matrix, $\overset{*}{QIB}$, to obtain the direction cosine matrix of the first, second, or third order update as a body to inertial frame transformation.

$$\overset{*}{DCM} = \overset{*}{QIB} * \overset{*}{DTM}$$

The direction cosine matrix, $\overset{*}{DCM}$, is normalized at intervals of DTNRM seconds or each time, $T \geq$ TNORM. Normalization takes the form of the first order orthonormalization.

$$\overset{*}{DCM} = \overset{*}{DCM} - \frac{1}{2} * \overset{*}{DCM} * (\overset{*}{DCM}^T * \overset{*}{DCM} - \overset{*}{I})$$

where $\overset{*}{I}$ is the identity matrix.

The normalization time is incremented

$$TNORM = T + DTNRM$$

The normalized direction cosine matrix is stored in the body to inertial frame transformation matrix, $\overset{*}{QIB}$, for use on the next pass.

$$\overset{*}{QIB} = \overset{*}{DCM}$$

Now form approximate midpoint direction cosine matrix, $\overset{*}{DCMMID}$, from $\overset{*}{DCM}$ of this and previous pass.

$$\overset{*}{DCMMID} = (\overset{*}{DCM} + \overset{*}{DCMOLD})/2$$

Using the approximate midpoint direction cosine matrix, $\overset{*}{DCMMID}$, obtained either from the quaternion or directio cosine matrix update, the compensated quantized integrals of the specific force, $\overline{DV}$, are transformed to the inertial frame

$$\overline{DVI} = \overset{*}{DCMMID} * \overline{DV}$$

The quantized specific force increments, $\overline{DVI}$, are summed to be compatible with the LLN module operating frequency. ($\overline{DVN}$ is zeroed in the LLN module) and the reference frame is transformed from the ALG inertial frame to the LLN inertial frame.

$$DVN(1) = DVN(1) + DVI(3)$$
$$DVN(2) = DVN(2) + DVI(1)$$
$$DVN(3) = DVN(3) + DVI(2)$$

The direction cosine matrix is converted from a body to inertial frame to a body to earth-fixed frame transformation.

$$\overset{*}{DCM} = \begin{vmatrix} SWET & 0.0 & CWET \\ CWET & 0.0 & -SWET \\ 0.0 & 1.0 & 0.0 \end{vmatrix} * \overset{*}{DCM}$$

where

$$CWET = COS(WE * T)$$

$$SWET = SIN(WE * T)$$

Finally if the update order is third order, NORDER = 3, the quantized integral of angular rate, $\overline{DTHETA}$ is stored in $\overline{DTHPRE}$ for use in the next pass

$$\overline{DTHPRE} = \overline{DTHETA}$$

The quantized integral of specific force, $\overline{DVN}$, the direction cosine matrix, $\overset{*}{DCM}$, and the quaternion, $\overline{QUAT}$ (if NQDCM = 0) are printed and the simulation time is incremented

$$TALG = T + DT$$

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6

On the initialization pass the title "ALGORITHM INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or NODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 of Vol. III for print control logic.

The printout is as follows:

Variable	Units	Description
$\overline{DVN}$	ft/s	quantized integral of specific force along the X,Y,Z accelerometer input axes in the inertial frame of the local level navigation module (LLN)
$\overline{QUAT}$	unity	quaternion
$\overset{*}{DCM}$	unity	the direction cosine matrix (to transform from the body frame to the earth fixed frame)

(7) <u>Subroutines Called</u> (See Vol. III, Section 2.3.15)

MTXM - Matrix transpose by matrix product

MXM - Matrix by matrix product

MXV - Matrix by vector product

2.3.13 LOCAL LEVEL NAVIGATOR MODULE (LLN)

(1) General Description

The software algorithms for a local-level wander-azimuth navigator are incorporated by this module. The inertially referenced incremental vehicle velocity, passed over from the attitude and velocity algorithm module (ALG), is transformed to local-level wander-azimuth coordinates, and then used to update the vehicle current position and velocity. The body-to-"new"-earth-fixed transformation from the ALG module, is further transformed to the "new" computational frame, and used in attitude computation. An altimeter loop is also employed, using the simulated baroaltimeter measurement from the altimeter module (ALTI), for the purpose of stabilizing the vertical channel.

(2) Local-Level Wander-Azimuth Navigator Module Computational Flow Diagram

The general flow logic of the LLN module is illustrated in Figure 1.

(3) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 80

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	0.02	s	module operating cycle time
2	PRNTSW	1.0	logical	print switch (0 - no print, otherwise print)
3	OUTSW	0.0	-	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	SPARE 1	0.0	-	not used
6	MODPOT	6.0	s	module print cycle time

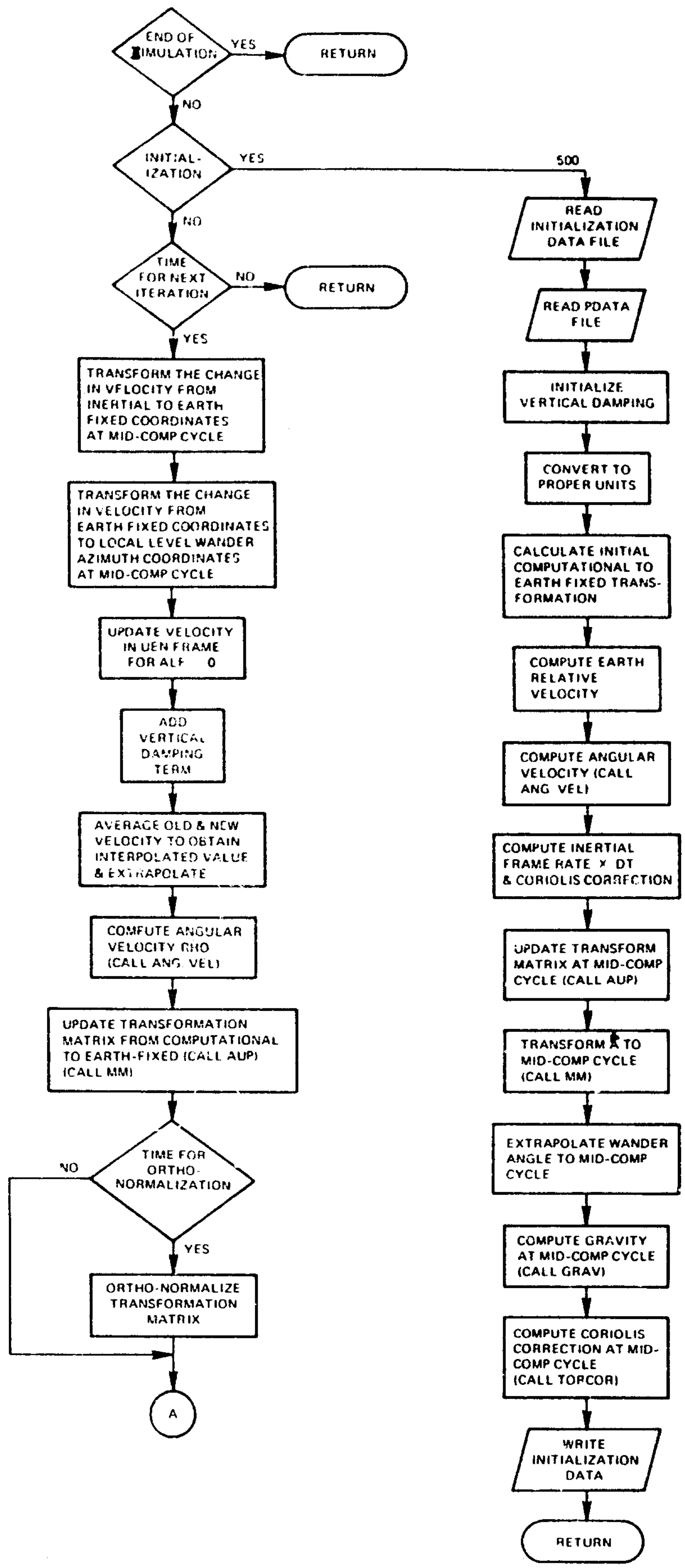

Figure 1. Local Level Navigator Module (Sheet 1 of 2)

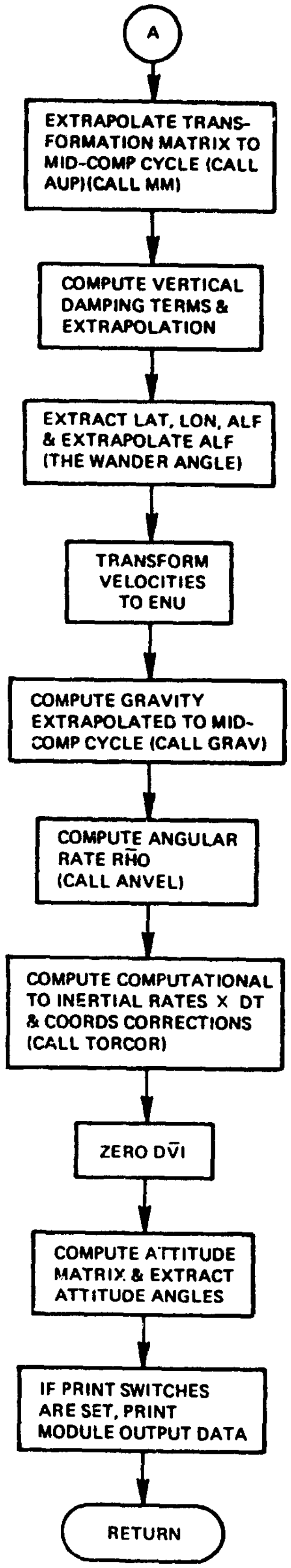

Figure 1. Local Level Navigator Module (Sheet 2 of 2)

Index	Variable	Default Value	Units	Description
7	ALTERR	0.0	ft	initial altitude error
8	VERR(1)	0.0	ft/s	initial velocity error (east)
9	VERR(2)	0.0	ft/s	initial velocity error (north)
10	VERR(3)	0.0	ft/s	initial velocity error (up)
11	LATERR	0.0	deg	initial latitude error
12	LONERR	0.0	deg	initial longitude error
13	CVD1	0.06	1/s	first vertical damping coefficient
14	CVD2	0.00162	$1/s^2$	second vertical damping coefficient
15	CVD3	0.0000162	$1/s^3$	third vertical damping coefficient

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.729211514E-4	rad/s	earth rotation rate
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s^2	nominal gravity
4	PRNTDT	6.0	s	printing frequency
5	ILAT	0.0	rad	initial latitude
6	ILON	0.0	rad	initial longitude
7	WONDER	0.0	rad	initial wander angle
8	IH	0.0	ft	initial altitude above sea level

Index	Variable	Default Value	Units	Description
9	ROLL	0.0	rad	initial roll angle
10	PITCH	0.0	rad	initial pitch angle
11	YAW	0.0	rad	initial yaw angle
12	DROLL	0.0	rad/s	first-time derivative of roll
13	DPITCH	0.0	rad/s	first-time derivative of pitch
14	DYAW	0.0	rad/s	first-time derivative of yaw
15	IV(1)	0.0	ft/s	initial velocity (VX) - EAST
16	IV(2)	0.0	ft/s	initial velocity (VY) - NORTH
17	IV(3)	0.0	ft/s	initial velocity (VZ) - UP
18	IAX	0.0	ft/s^2	initial specific force in body frame (AX) - EAST
19	IAY	0.0	ft/s^2	initial specific force in body frame (AY) - NORTH
20	IAZ	0.0	ft/s^2	initial specific force in body frame (AZ) - UP

(4) Call-Line Data

$$\overbrace{\underbrace{\text{T, IENDF, DVI, ALTO, DCM,}}_{\text{OUT}}}^{\text{IN}}$$

$$\underbrace{\text{NAVLAT, NAVLON, NAVV, NAVH, NAVP, NAVR, NAVHD)}}_{\text{OUT}}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	INTEGER	last-pass indicator
$\overline{DVI}$	ft/s	REAL	quantized integral of vehicle specific force in the inertial frame
ALTO	ft	REAL	indicated altitude from baroaltimeter
$\overset{*}{DCM}$	unity	REAL	direction cosine matrix (body to earth-fixed frame transformation)

(b) Call-Line Output

Variable	Units	Data Type	Description
NAVLAT	deg	REAL	computed navigational latitude
NAVLON	deg	REAL	computed navigational longitude
$\overline{NAVV}$	ft/s	REAL	computed navigational velocity vector in ENU frame
NAVH	ft	REAL	computed navigational altitude
NAVP	deg	REAL	computed navigational pitch
NAVR	deg	REAL	computed navigational roll
NAVHD	deg	REAL	computed navigational heading

(5) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE).
- Set OFILE = XFILE.

Initialize for vertical damping.

H = IH + ALTERR

HB = ALTO

OHB = ALTO

DELH = HB - H

VDMP = CVD3 * DELH * DT

NUP = 0

NORTH = 4

Add errors and convert units of latitude and longitude to radians

INAVLA = ILAT + (LATERR/RDTODG)

INAVLO = ILON + (LONERR/RDTODB)

where RDTODG = 57.29577951 deg/rad

The initial up, east, north (UEN) wander angle to earth-fixed transformation matrix, $\overset{*}{A}$, is computed and printed.

$$\overset{*}{A} = \left[\begin{array}{c|c|c} \text{COSLT*COSLON} & \text{-CALF*SINLON - SALF*SINLT*COSLON} & \text{SALF*SINLON - CALF*SINLT*COSLON} \\ \text{COSLT*SINLON} & \text{COSLON*CALF - SALF*SINLT*SINLON} & \text{-SALF*COSLON - CALF*SINLT*SINLON} \\ \text{SINLT} & \text{SALF*COSLT} & \text{COSLT*CALF} \end{array}\right]$$

where

SINLT = SIN (INAVLA)

COSLT = COS (INAVLA)

SINLON = SIN (INAVLO)

COSLON = COS (INAVLO)

SALF = SIN (WONDER)

CALF = COS (WONDER)

Add velocity errors to velocity

$$\overline{NAVV} = \overline{IV} + \overline{VERR}$$

and compute initial earth-relative velocity in UEN - wander-angle frame.

V(1) = NAVV(3)

V(2) = CALF * NAVV(1) + SALF * NAVV(2)

V(3) = -SALF*NAVV(1) + CALF * NAVV(2)

Compute the mid cycle altitude, XH

XH = H + 0.5 * V(1) * DT

Compute the initial angular velocity, $\overline{RHO}$, of the UEN wander angle earth-fixed frame in the UEN-wander angle frame.

Set RHO(1) = 0.0

Call angular velocity computational subroutine (Section 7b)

CALL ANGVEL ($\overline{V}$, CALF, SSLAT, H, $\overline{RHO}$)

where the input variables are

$\overline{V}$ = initial earth relative velocity vector

CALF = COSINE of the wander angle

SALF = SINE of the wander angle

SSLAT = SINE of the latitude squared

H = altitude

and the quantity output is

$\overline{RHO}$ = angular velocity of the UEN-wander angle earth-fixed frame.

(See Section 7b for subroutine ANGVEL description.)

Compute computational to inertial frame rate by DT and coriolis corrections.

Call TORCOR ($\overset{*}{A}$, $\overline{RHO}$, $\overline{V}$, DT, $\overline{THG}$, $\overline{WXV}$)

where the input variables are

$\overset{*}{A}$ = UEN-wander angle to earth-fixed transformation

$\overline{RHO}$ = initial angular velocity in UEN WA frame

$\overline{V}$ = initial earth relative velocity

DT = module operating cycle

and the output variables are

$\overline{THG}$ = computational frame with respect to inertial frame rate times DT.

$\overline{WXV}$ = coriolis correction

(See Section 7c for subroutine TORCOR description.)

To extrapolate $\overset{*}{A}$ to mid computation cycle and restore as $\overset{*}{X}A$ first define THY and THZ, the Y and Z components of the angular rate times DT at mid cycle.

THY = RHO(2) * DT/2

THZ = RHO(3) * DT/2

Form the update matrix, $M2\overset{*}{D}T$

CALL AUP (THY, THZ, $M2\overset{*}{D}T$)

and

$$\text{CALL MM } (\overset{*}{A}, \overset{*}{M2DT}, \overset{*}{XA})$$

to update the transformation matrix, $\overset{*}{A}$, to $\overset{*}{XA}$ matrix. (See Section 7d for subroutine AUP description).

Extrapolate the wander angle, ALF, to mid computation cycle

DALF = (V(2) * SINLT * DT/2)/(RE * COSLT)

where

V(2) = east velocity component
SINLT = SINE of the latitude
DT = module operating cycle
RE = earth radius
COSLT = COSINE of the latitude

and calculate

XSALF = SALF + DALF * CALF
XCALF = CALF - DALF * SALF

Compute gravity at mid computation cycle

$$\text{CALL GRAV } (\overset{*}{XA}, XH, XSSLAT, \overline{GR})$$

where the input variables are

$\overset{*}{XA}$ = mid computation cycle UEN-WA to earth-fixed transformation matrix
XH = the mid computation cycle altitude

and the output variables are

XSSLAT = SINE of the latitude squared at mid computation cycle

$\overline{GR}$ = gravity vector at mid computation cycle in the computational frame

(See Section 7e for subroutine GRAV description).

Compute the coriolis correction, WXV, at mid computation cycle.

$$\text{CALL TORCOR } (\overset{*}{XA}, \overline{RHO}, \overline{V}, DT, \overline{THG}, \overline{WXV})$$

where the quantities are described above.

The quantized integral of specific force, $\overline{DVI}$, is initialized for the accumulation at the algorithm (ALG) module cycle time.

$$\overline{DVI} = (0.0, 0.0, 0.0)$$

The units of initial latitude, ILAT, and longitude, ILON, are converted from radians to degrees and restored as

ITEMP1 = ILAT * RDTODG

ITEMP2 = ILON * RDTODG

The initialization switch is reset,

$$INITSW = 1$$

and the simulation time incremented

$$TLLN = T + DT$$

The subroutine returns to the main program.

(b) General Function

The following functions are performed every module operating cycle.

The quantized integral of specific force, $\overline{DVI}$, is transformed from inertial to earth-fixed coordinates at the mid-computation cycle.

or

$$\overline{DVE} = Z(WET) * \overline{DVI}$$

$$\overline{DVE} = \begin{vmatrix} CWET & SWET & 0 \\ -SWET & CWET & 0 \\ 0 & 0 & 1 \end{vmatrix} * \overline{DVI}$$

where

$$CWET = COS(WET)$$

$$SWET = SIN(WET)$$

and

$$WET = WE * (T - DT/2)$$

and $\overline{DVE}$ is further transformed to $\overline{DVC}$ or from earth-fixed to computational frame at mid-computation cycle.

$$\overline{DVC} = \overline{DVE} * \overset{*}{XA}$$

The velocity from the previous pass is restored as $\overline{OV}$.

$$\overline{OV} = \overline{V}$$

The velocity, $\overline{V}$, is now updated in the UEN frame for the wander angle = 0.

$$\overline{V} = \overline{OV} + \overline{DVC} + \overline{WXV} + \overline{GR} * DT$$

and a vertical damping term is added

$$\overline{V} = \overline{V} + (VDMP + CVD2 * DELH) * DT$$

The velocity and its value on the previous pass are averaged to obtain an interpolated value, $\overline{HV}$

$$\overline{HV} = (\overline{V} + \overline{OV})/2$$

and extrapolated to obtain $\overline{XV}$

$$\overline{XV} = (3 * \overline{V} - \overline{OV})/2$$

To compute the angular velocity, RHO, the subroutine ANGVEL is called, (see Section 7b).

CALL ANGVEL ($\overline{HV}$, XCALF, XSALF, XSSLAT, XH, $\overline{RHO}$)

where the arrangements are

$\overline{HV}$	=	the average velocity
XSALF	=	SINE of the mid-computation cycle wander angle
XCALF	=	COSINE of the mid-computation cycle wander angle
ASSLAT	=	SINE of the latitude squared
XH	=	the mid-computation cycle altitude
$\overline{RHO}$	=	the angular velocity

Using the angular velocity, RHO, the computational to earth fixed transformation matrix, $\overset{*}{A}$, is calculated.

First, THY and THZ are defined,

$$THY = RHO(2) * DT$$

$$THZ = RHO(3) * DT$$

and the $\overset{*}{A}$ matrix update subroutine is called to obtain the matrix, $\overset{*}{M2DT}$

CALL AUP (THY, THZ, $\overset{*}{M2DT}$)

The matrix multiplication of the old value of the $\overset{*}{A}$ - matrix is multiplied by $\overset{*}{M2DT}$ to obtain the matrix, $\overset{*}{ATEMP}$

$$\overset{*}{ATEMP} = \overset{*}{A} * \overset{*}{M2DT}$$

and $\overset{*}{ATEMP}$ is restored as the updated matrix $\overset{*}{A}$

$$\overset{*}{A} = \overset{*}{ATEMP}$$

The update counter initially set to zero is incremented

$$NUP = NUP + 1$$

The orthonormalization of the transformation matrix, $\overset{*}{A}$, is by passed, when the update counter, NUP, is less than the orthonormalization parameter, NORTH

IF (NUP .LT. NORTH)

On an orthonormalization pass, reset

$$NUP = 0$$

and the $\overset{*}{A}$ transformation matrix is now orthonormalized

$$\overset{*}{A} = \overset{*}{A} - 1/2 * \overset{*}{A} * (\overset{*}{A}^{T}\overset{*}{A} - \overset{*}{I})$$

and $\overset{*}{AV}$ set equal to $\overset{*}{A}$ to obtain the $\overset{*}{A}$ matrix in row vector form.

$$\overset{*}{AV} = \overset{*}{A}$$

The extrapolated mid-computation cycle altitude error is calculated,

$$DELH = XHB - XH$$

from the error between the extrapolated barometric and the computed altitude.

The vertical acceleration error, VDMP, is

$$VDMP = VDMP + CVD3 * DELH * DT$$

The computed altitude, H, is stored as the variable NAVH

$$NAVH = H$$

The computed latitude, NAVLAT,

$$NAVLAT = ARCTAN(A(3,1)/(A(1,1)^2 + A(2,1)^2)^{1/2})$$

the computed longitude, NAVLON,

$$NAVLON = ARCTAN(A(2,1)/A(1,1))$$

and the wander angle, ALF,

$$ALF = ARCTAN(A(3,2)/A(3,3))$$

The wander angle from the previous pass is denoted as,

$$OALF = ALF$$

and the mid-computation cycle value of the wander angle is,

$$DALF = (OALF + ALF)/2$$

From the wander angle, ALF, are calculated,

$$CALF = COS(ALF)$$

$$SALF = SIN(ALF)$$

Now the transformation matrix, $\overset{*}{A}$, is extrapolated to mid-computation cycle. THY and THZ are set equal to half of their respective value.

THY = THY/2

THZ = THZ/2

and the $\overset{*}{A}$ matrix update subroutine is called to obtain the $\text{M2}\overset{*}{\text{D}}\text{T02}$ matrix. (Section 7d)

```
CALL AUP (THY, THZ, M2DT02)
```

The updated transformation matrix, $\text{X}\overset{*}{\text{A}}$, is formed in the matrix multiplication,

$$\overset{*}{\text{XA}} = \overset{*}{\text{A}} \ * \ \text{M2}\overset{*}{\text{D}}\text{T02}$$

An updated altitude, H, is calculated from the mid-computations cycle vertical velocity and the error in the extrapolated values of the calculated altitude and the barometric altitude.

H = H + (HV(1) + CVD1 * DELH) * DT

An extrapolated altitude XH is calculated

XH = H + V(1) * DT/2

The barometric altitude, ALTO is redefined as HB

HB = ALTO

and the extrapolated barometric altitude is calculated

XHB = (3*HB - OHB)/2

where

OHB = HB, the barometric altitude on the previous pass and the extrapolated values of CALF and SALF are calculated from

$$XSALF = SALF + DALF * CALF$$

$$XCALF = CALF - DALF * SALF$$

and the computed velocity vector, NAVV, is tranformed to ENU fram or

$$\overline{NAVV} = Z(-ALF) * \overline{V}$$

$$\overline{NAVV} = \begin{vmatrix} CALF & -SALF & 0 \\ SALF & CALF & 0 \\ 0 & 0 & 1 \end{vmatrix} * \overline{V}$$

Gravity, GR, extrapolated to mid-computation cycle is calculated by the subroutine,

CALL GRAV ($\overset{*}{XA}$, XH, XSSLAT, GR)

where $\overset{*}{XA}$ = the mid-computation cycle extrapolated transformation matrix

XH = The mid-computation cycle extrapolated altitude

XSSLAT = The mid-computation cycle extrapolated squared sine of the latitude

GR = the gravity vector at mid comp cycle.

The angular velocity, $\overline{RHO}$, is calculated by the subroutine

CALL ANGVEL ($\overline{XV}$, XCALF, XSALF, XSSLAT, XH, $\overline{RHO}$)

where $\overline{XV}$, XCALF, XSALF, XSSLAT, and XH are quantities extrapolated to mid-computation cycle.

Using the angular velocity, $\overline{RHO}$, the subroutine TORCOR is called to calculate the vector, $\overline{THG}$, which defines the computation to inertial frame rates times DT., and the coriolis corrections, $\overline{WXV}$, at mid-computation cycle.

CALL TORCOR ($\overset{*}{XA}$, $\overline{RHO}$, $\overline{XV}$, DT, $\overline{THG}$, $\overline{WXV}$)

The quantized integral of specific force, $D\overline{V}I$, is reinitialized for the accumulation at the algorithm (ALG) module cycle time.

$$D\overline{V}I = (0.0,\ 0.0,\ 0.0)$$

The body to computation frame transformation (attitude) matrix, $\overset{*}{DTEM}$, is calculated

$$\overset{*}{DTEM} = \overset{*}{AV}{}^{T} * \overset{*}{DCM}$$

The computed attitude angles are extracted. First the computed pitch angle, NAVP

$$NAVP = ARCTAN(DTEM(1)/DEN)$$

where

$$DEN = (DTEM(2)^2 + DTEM(3)^2)^{1/2}$$

The computed roll angle, NAVR,

$$NAVR = ARCTAN(DTEM(2)/DTEM(3))$$

and the computed heading, NAVHD,

$$NAVHD = ARCTAN(DTEM(4)/DTEM(7)) - ALF$$

If DEN = 0.0, the pitch is ±90°, the previous attitude values are output, the simulation time is incremented, TLLN = T + DT, and the subroutine returns to the main program.

Otherwise the computed attitude and position values are converted from radians to degrees.

$$ITEMP1 = NAVLAT * RDTODG$$

$$ITEMP2 = NAVLON * RDTODG$$

$$ITEMP3 = NAVP * RDTODG$$

$$ITEMP4 = NAVR * RDTODG$$

$$ITEMP5 = NAVHD * RDTODG$$

and NAVH, $NA\overline{V}V$, ITEMP1, ITEMP2, ITEMP3, ITEMP4, ITEMP5, $\overset{*}{A}$ and $DT\overset{*}{E}M$ are printed, and the simulation time incremented

$$TLLN = T + DT$$

The subroutine returns to the main program.

(6) Output

(a) Print

FORTRAN unit number: OFILE = 6.

On the initialization pass the title "NAVIGATION INITIALIZATION" and the initialization data are printed.

Printed output is produced at PRNTDT or MODPDT intervals when PRNTSW $\geq$ 1. See Section 2.2 for print control logic. The printed output is as follows,

Variable	Units	Description
NAVH	ft	computed navigational altitude
$NA\overline{V}V$	ft/s	computed navigational velocity vector in ENU frame
NAVLAT	deg	computed navigational latitude
NAVLON	deg	computed navigational longitude
NAVP	deg	computed navigational pitch
NAVR	deg	computed navigational roll

Variable	Units	Description
NAVHD	deg	computed navigational heading
$\overset{*}{\text{A}}$	unity	the UP, east, north wander angle (Computational) frame to earth-fixed transformation matrix
$\overset{*}{\text{DTEM}}$	unity	the body to computation frame (attitude) transformation matrix.

(7) Subroutines Called

(a) MTXM = matrix transpose by matrix multiplication. (See Vol. III Section 2.3.15)

(b) Angular Velocity (ANGVEL)

This subroutine computes level components of the vehicle angular velocity due to its motion with respect to the earth.

Call Line Data

$$\overbrace{\overline{\text{AVEL}},\ \text{ACALF},\ \text{ASALF},\ \text{ASSL},\ \text{ALT},}^{\text{INPUT}}\ \underbrace{\overline{\text{RHO}}}_{\text{Output}}$$

Call Line Input

Variable	Data Type	Description
AVEL	REAL	earth relative velocity vector
ACALF	REAL	COSINE of the wander angle
ASALF	REAL	SINE of the wander angle
ASSL	REAL	SINE of the latitude aquared
ALT	REAL	altitude

Call Line Output

Variable	Data Type	Description
$\overline{RHO}$	REAL	angular velocity vector in UEN - WA frame

Formulation

Compute the east and north components of the earth relative velocity in the UEN - WA frame

$$VE = ACALF * AVEL(2) - ASALF * AVEL(3)$$
$$VN = ASALF * AVEL(2) + ACALF * AVEL(3)$$

Calculate the radii, RM and RP, of curuature of the WGS-72 earth model for the meridional plane (N-S) and the prime vertical plane (E-W), respectively,

$$RP = ALT + RE/(1-ESQ * ASSL)$$
$$RM = ALT + RESQ/(1-ESQ * ASSL)^{3/2}$$

where

RE = 2.0925640E7, the earth's radius.

ESQ = 0.006694317778, the square of the eccentricity of the earth's meridional ellipse.

RESQ = 2.078555712E7, = RE * (1-ESQ)

Now the east and north components of the angular rate, WE, and WN, are calculated

$$WE = -VN/RM$$
$$WN = VE/RP$$

and are converted to the angular velocity, $\overline{RHO}$, of the UEN-WA frame with respect to the earth-fixed frame.

RHO(1) = 0.0

RHO(2) = ACALF * WE + ASALF * WN

RHO(3) = -ASALF * WE + ACALF * WN

(See flow chart, Figure 2).

(c) Angular Velocity by DT and Coriols Corrections (TORCOR)

This subroutine computes computational to inertial frame angular velocities times DT and the coriolis corrections.

Call Line Data

$$\overbrace{\overset{*}{A},\ \overline{RHO},\ \overline{V},\ DT,}^{\text{INPUT}}\qquad \underbrace{\overline{THG},\ \overline{WXV}}$$

Call Line Input

Variable	Data Type	Description
$\overset{*}{A}$	REAL	UEN-WA earth-fixed transformation
$\overline{RHO}$	REAL	angular velocity vector in UEN-WA frame
$\overline{V}$	REAL	earth relative velocity vector
DT	REAL	module operating cycle

Call Line Output

Variable	Data Type	Description
$\overline{THG}$	REAL	Computational frame with respect to inertial frame angular rate times DT
$\overline{WXV}$	REAL	coriolis correction

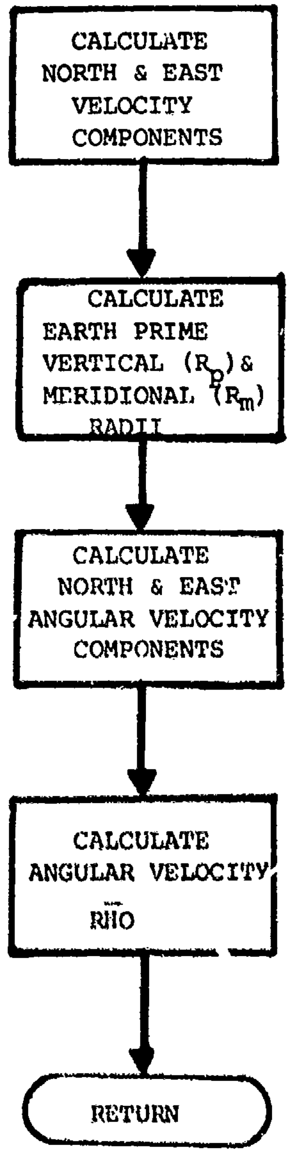

Figure 2. Angular velocity computation module (ANGVEL)

Formulation

Calculate the earth rate signals, $\overline{EAR}$, in the computational frame.

$$EAR(1) = A(3,1) * WE * DT$$
$$EAR(2) = A(3,2) * WE * DT$$
$$EAR(3) = A(3,3) * WE * DT$$

where WE = 0.7292115147E-5 rad/sec, the earth rotation rate.

The gyro torquing signals, THG, are

$$\overline{THG} = \overline{EAR} + \overline{RHO} * DT$$

and the angular velocity (for $\overline{WXV}$), $\overline{THE}$,

$$\overline{THE} = \overline{THG} + \overline{EAR}$$

Compute the vector cross product of the angular rate and the vehicle velocity to obtain the coriolis correction, $\overline{WXV}$, for the velocity update.

$$\overline{WXV} = \overline{THE} \times \overline{V}$$

(See flow chart, Figure 3).

(d) DCM Second Order Update Matrix (AUP)

This subroutine forms the second order update matrix for the local vertical-wander angle computational to earth fixed frame direction cosine matrix (DCM).

Call Line Data

INPUT

DY, DZ,

$\overset{*}{\text{MUP}}$

OUTPUT

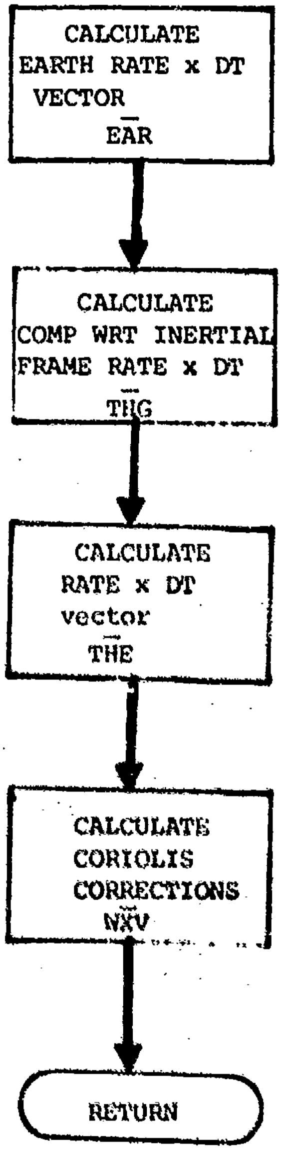

Figure 3. Angular velocity times DT and Coriolis Correction Module (TORCOR)

Call Line Input

Variable	Data Type	Description
DY	REAL	Y-axis increment
DZ	REAL	Z-axis increment

Call Line Output

Variable	Data Type	Description
$\overset{*}{MUP}$	REAL	Second order update matrix

Formulation

Calculate the second order update matrix, $\overset{*}{MUP}$

$$\overset{*}{MUP} = \begin{vmatrix} 1-(DY^2 + DZ^2)/2 & -DZ & DY \\ DZ & 1-DZ^2/2 & DY * DZ/2 \\ -DY & DY * DZ/2 & 1-DY^2/2 \end{vmatrix}$$

(See flow chart, Figure 4).

(e) Gravity Computation (GRAV)

This subroutine computes three components of gravity in the computational frame using the WGS-72 ellipsoidal earth model.

Call Line Data

INPUT

$\overset{*}{AM}$, ALT,

SSL, $\bar{G}$

OUTPUT

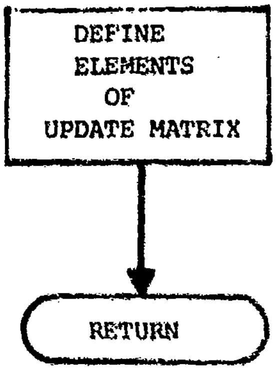

Figure 4. Update for computation to earth fixed transformation matrix module (AUP)

Call Line Input

Variable	Data Type	Description
AM^*	REAL	computational frame transformation matrix
ALT	REAL	altitude

Call Line Output

Variable	Data Type	Description
SSL	REAL	AM(3,1) squared
$\overline{G}$	REAL	gravity vector

Formulation

Define $\quad SSL = AM(3,1)^2$

and $\quad COEF = 1.63E\text{-}8 * ALT * AM(3,1)$

Calculate the three components of gravity, $\overline{G}$, in the computational frame.

$$G(1) = -(32.0877057 + 0.16939081 * SSL + 7.5281E\text{-}4 * SSL^2)$$
$$*(1-(9.6227E-8-S-5-6.\ 408E\text{-}10 * SSL) * ALT$$
$$+ 6.8512E\text{-}15 * ALT^2)$$

$$G(2) = COEF * AM\ (3,2)$$
$$G(3) = COEF * AM\ (3,3)$$

(See flow chart, Figure 5).

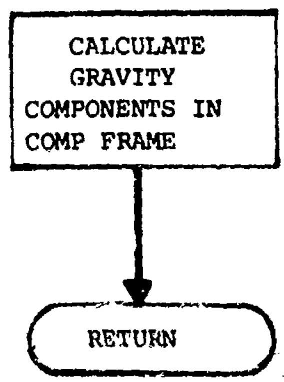

Figure 5. Gravity Computation Module (GRAV)

(f) Matrix Multiply (MM)

This subroutine forms the product of two 3x3 matrices.

Call Line Data

$$\overbrace{\overset{*}{A}\ ,\ \overset{*}{B},}^{\text{INPUT}}\ \underbrace{\overset{*}{C}}_{\text{OUTPUT}}$$

Call Line Input

Variable	Data Type	Description
$\overset{*}{A}$	REAL	first matrix
$\overset{*}{B}$	REAL	second matrix

Call Line Output

Variable	Data Type	Description
$\overset{*}{C}$	REAL	product of input matrices

Formulation

Compute matrix products, $\overset{*}{C}$

$$\overset{*}{C} = \overset{*}{A} * \overset{*}{B}$$

(See flow chart, Figure 6).

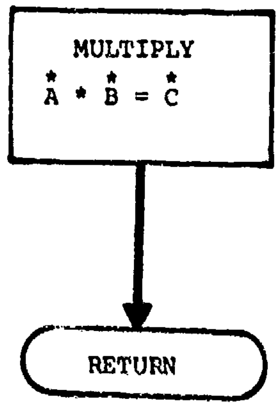

Figure 6. Matrix Multiply Module (MM)

2.3.14 EVALUATION MODULE (EVL)

(1) General Description

This module was included to provide the user with a means of performing an error analysis of—or otherwise manipulating—the INSS trajectory and navigation data during a simulation, or collecting simulation data for post-run analysis. The EVL module simply computes and prints a table of navigation and attitude errors and trajectory parameters every 50 module-operating cycles. The printing frequency is based on the user-specified operating-frequency time (DT).

(2) Input

(a) Module Initialization File (IFILE)

FORTRAN unit number: 90

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	DT	1.0	s	module operating cycle time
2	PRNTSW	1.0	logical	print switch (0-noprint, otherwise print)
3	OUTSW	0.0	-	not used
4	XFILE	6.0	logical	FORTRAN unit number for printout
5	PRNTDT	1.0	s	printing cycle time

(b) Common Initialization File (PFILE)

FORTRAN unit number: 7

FORTRAN format: I5, F20.10

Index	Variable	Default Value	Units	Description
1	WE	0.7292115147E-4	rad/s	earth rotation ra
2	RE	20925640.0	ft	earth radius
3	G	32.2	ft/s	nominal gravity

(3) Call-Line Data

$$\overbrace{\mathrm{T,\ IENDF,\ LAT,\ LON,\ ALT,\ \overline{VEL},\ \overline{DVT},\ PITCH,\ ROLL,\ YAW,\ WONDER,\ NAVLAT,}}^{\mathrm{INPUT}}$$

$$\overbrace{\mathrm{NAVLON,\ NA\overline{V}V,\ NAVH,\ NAVP,\ NAVR,\ NAVHD}}^{\mathrm{INPUT}}\mathrm{)}$$

(a) Call-Line Input

Variable	Units	Data Type	Description
T	s	REAL	current simulation time
IENDF	logical	REAL	last-pass indicator
LAT	rad	REAL	geodetic latitude - "true"
LON	rad	REAL	geodetic longitude - "true"
ALT	ft	REAL	altitude above sea level - "true"
$\overline{\mathrm{VEL}}$	ft/s	REAL	velocity vector in body frame - "true"
$\overline{\mathrm{DVT}}$	ft/s	REAL	quantized integral of specific force in ENU frame - "true"
PITCH	rad	REAL	vehicle pitch - "true"
ROLL	rad	REAL	vehicle roll - "true"
YAW	rad	REAL	vehicle yaw - "true"
WONDER	rad	REAL	vehicle wander angle - "true" (clockwise from north about the local vertical up axis)
NAVLAT	rad	REAL	computed navigational latitude
NAVLON	rad	REAL	computed navigational longitude
NAVH	ft	REAL	computed navigational altitude

Variable	Units	Data Type	Description
NAVP	rad	REAL	computed navigational pitch
NAVR	rad	REAL	computed navigational roll
NAVHD	rad	REAL	computed navigational heading

(b) Call-Line Output

There are no output parameters in the call-line argument list.

(4) Formulation

(a) Initialization Function

The initialization switch, INITSW, is initialized to zero in a DATA statement. The following functions are performed on the first pass, if INITSW = 0, IENDF = 0, and if the module operating cycle time has elapsed.

- Read and print initialization file (IFILE).
- Read common initialization file (PFILE)
- Set OFILE = XFILE.

The initialization switch is set, INITSW = 1, and the current simulation time is incremented,

$$TEVL = T + DT$$

Now the subroutine returns to the main program.

(b) General Function

The following functions are performed every module operating cycle.

The 13 trajectory and the 9 navigation error parameters are arrayed in 50-element vector form and the unit converted to units convenient for interpretation.

XT(N)	=	T	(s)
XLAT(N)	=	LAT * RDTODG	(rad to deg)
XLON(N)	=	LON * RDTODG	(rad to deg)
XALT(N)	=	ALT	(ft)
XVELX(N)	=	VEL(1)	(ft/s)
XVELY(N)	=	VEL(2)	(ft/s)
XVELZ(N)	=	VEL(3)	(ft/s)
XDVTX(N)	=	DVT(1)	(ft/s)
XDVTY(N)	=	DVT(2)	(ft/s)
XDVTZ(N)	=	DVT(3)	(ft/s)
XHEAD(N)	=	(YAW-WONDER) * RDTODG	(rad to deg)
XPITCH(N)	=	PITCH * RDTODG	(rad to deg)
XROLL(N)	=	ROLL * RDTODG	(rad to deg)

The navigation errors are the differences between the "true" position and attitude values and the corresponding computed navigational values in their appropriate units.

ELAT(N)	=	(LAT-NAVLAT)*RE	(rad to ft)
ELON(N)	=	(LON-NAVLON)*RE*COS(LAT)	(rad to ft)
EALT(N)	=	ALT - NAVH	(ft)
EVELX(N)	=	VEL(1) - NAVV(1)	(ft/s)
EVELY(N)	=	VEL(2) - NAVV(2)	(ft/s)
EVELZ(N)	=	VEL(3) - NAVV(3)	(ft/s)
EHEAD(N)	=	(YAW - WONDER - NAVHD) *3600*RDTODG	(rad to sec)
EPITCH(N)	=	(PITCH - NAV)*3600*RDTODG	(rad to sec)
EROLL(N)	=	(ROLL - NAVR)*3600*RDTODG	(rad to sec)

The listed trajectory and navigation error parameters are computed and stored for N = 50 module operating cycles. When the 50 lines of trajectory and the 50 lines of navigational error values are collected, each of these pages of data is then printed and written on an unformatted tape or disc file.

The simulation time is incremented

$$TEVL = T + DT$$

The subroutine now returns to the main program.

(5) Output

(a) Print

FORTRAN unit number: OFILE = 6.

On the initialization pass the title "EVALUATION INITIALIZATION" and the initialization data are printed.

A printed output table is produced every 50 module operating cycles or on last pass. The remaining data is printed, if the print switch, PRNTSW $\geq$ 1. A sample of the printed output is displayed in Section 2.2. The printed output consists of the following list of variables.

Variable	Units	Description
XT	s	current simulation time
XLAT	deg	geodetic latitude - "true"
XLON	deg	geodetic longitude - "true"
XALT	ft	altitude above sea level - "true"
XVELX	ft/s	east velocity component - "true"
XVELY	ft/s	north velocity component - "true"
XVELZ	ft/s	up velocity component - "true"
XHEAD	deg	heading - "true"

Variable	Units	Description
XP$\overline{\text{I}}$TCH	deg	pitch - "true"
XR$\overline{\text{O}}$LL	deg	roll - "true"
EL$\overline{\text{A}}$T	ft	latitude error ("true" - computed)
EL$\overline{\text{O}}$N	ft	longitude error ("true" - computed)
E$\overline{\text{A}}$LT	ft/s	altitude error ("true" - computed)
EV$\overline{\text{E}}$LX	ft/s	east velocity error ("true" - computed)
EV$\overline{\text{E}}$LY	ft/s	north velocity error ("true" - computed)
EV$\overline{\text{E}}$LZ	ft/s	up velocity error ("true" - computed)
EH$\overline{\text{E}}$AD	s	heading error ("true" - computed)
EP$\overline{\text{I}}$TCH	s	pitch error ("true" - computed)
ER$\overline{\text{O}}$LL	s	roll error ("true" - computed)

(b) Secondary Storage (PPFILE)

FORTRAN unit number: 12

FORTRAN format: unformatted

The following output is written on tape or disc on every module cycle for later postprocessing.

The variable list is identical with the printed list above.

(7) Subroutines Called

No subroutines are called.

2.3.15 MATHEMATICAL SUBROUTINES

(1) General Description

The INSS sytem includes a library of three dimensional mathematical subroutines that perform vector-matrix operations and random number generation. All vectors must be dimensioned V(3). All matrices must be dimensioned M(9) and stored sequentially by rows, i.e.,

$$\overset{*}{M} = \begin{vmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{vmatrix} = \begin{vmatrix} M(1) & M(2) & M(3) \\ M(4) & M(5) & M(6) \\ M(7) & M(8) & M(9) \end{vmatrix}$$

(2) Subroutines

The following subroutines are described.

(a) Matrix Multiplication (MXM)

This subroutine multiplies two 9-element matrices.

Call Line Data

$$(\overbrace{\overset{*}{M}1,\ \overset{*}{M}2,}^{\text{INPUT}}\ \underbrace{\overset{*}{M}3}_{\text{OUTPUT}})$$

Call Line Input

Variable	Data Type	Description
$\overset{*}{M}1$	REAL	A 3 x 3 matrix stored by row as a 9-element vector
$\overset{*}{M}2$	REAL	A 3 x 3 matrix stored by row as a 9-element vector

Call Line Output

Variable	Data Type	Description
$\overset{*}{M}3$	REAL	A 3 x 3 matrix stored by row as a 9-element vector

Formulation

The matrices $\overset{*}{M}1$ and $\overset{*}{M}2$ are multiplied and stored in matrix $\overset{*}{M}3$

$$\overset{*}{M}3 = \overset{*}{M}1 * \overset{*}{M}2$$

(b) Matrix by Vector Product (MXV)

A matrix time vector product is produced to obtain a vector.

Call Line Data

$$(\overbrace{\overset{*}{M}, \overline{V1},}^{\text{INPUT}} \; \underbrace{\overline{V2})}_{\text{OUTPUT}}$$

Call Line Input

Variable	Data Type	Description
$\overset{*}{M}$	REAL	A 3 x 3 matrix stored by now as nine element vector
$\overline{V1}$	REAL	A 3 element vector

Call Line Output

Variable	Data Type	Description
$\overline{V2}$	REAL	A 3 element vector

Formulation

The matrix, $\overset{*}{M}$, is multiplied by the vector, $\overline{V1}$ and stored as the vector, $\overline{V2}$.

$$\overline{V2} = \overset{*}{M} * \overline{V1}$$

(c) Matrix Transpose by Matrix Product (MTXM)

A transposed matrix is multiplied by another matrix to obtain a third matrix.

Call Line Data

$$(\underbrace{\overset{*}{M1},\ \overset{*}{M2}}_{\text{INPUT}},\ \underbrace{\overset{*}{M3}}_{\text{OUTPUT}})$$

Call Line Input

Variable	Data Type	Description
$\overset{*}{M1}$	REAL	A 3 x 3 matrix stored by row as a nine element vector
$\overset{*}{M2}$	REAL	A 3 x 3 matrix stored by row as a nine element vector

Call Line Output

Variable	Data Type	Description
$\overset{*}{M3}$	REAL	A 3 x 3 matrix stored by row as a nine element vector

Formulation

The transpose of matrix, $\overset{*}{M1}$, is multiplied by the matrix, $\overset{*}{M2}$, and stored as the matrix, $\overset{*}{M3}$.

$$\overset{*}{M3} = \overset{*}{M1}^T * \overset{*}{M2}$$

(d) Matrix Transpose by Vector Product (MTXV)

A transpose matrix is multiplied by a vector to obtain a vector.

Call Line Data

$$(\underbrace{\overset{*}{M}, \overline{V1}}_{\text{INPUT}}, \underbrace{\overline{V2}}_{\text{OUTPUT}})$$

Call Line Input

Variable	Data Type	Description
$\overset{*}{M}$	REAL	A 3 x 3 matrix stored bv row as a nine element vector
$\overline{V1}$	REAL	A 3 element vector

Call Line Output

Variable	Data Type	Description
$\overline{V2}$	REAL	A 3 element vector

Formulation

A transpose matrix, $\overset{*}{M}$, is multiplied by a vector, $\overline{V1}$, and stored as the vector, $\overline{V2}$.

$$\overline{V2} = \overset{*}{M}^T * \overline{V1}$$

(e) Random Number Generator (GAUSS)

This function computes a Gaussian random number with a given mean and standard deviation. It is machine independent and will work properly if the single precision word of the machine is more than 20 bits long.

Call Line Data

$$\overbrace{\text{(MEAN,STD)}}^{\text{INPUT}}$$

Call Line Input

Variable	Data Type	Description
STD	REAL	desired standard deviation of the normal distribution
MEAN	REAL	desired mean of normal distribution

Formulation

An odd integer seed is selected, initially, as IX = 3. A real number, FA, is initialized to zero.

Twelve uniformly distributed numbers, FY, from 0 to +1 are calculated.

Those numbers are summed to obtain the real gaussian random number, FA, which has a mean of 6 and standard deviation of 1.0.

The real random number GAUSS, returned by the function is from a Gaussian dstribution having a mean value of MEAN and a standard deviation of STD. It is calculated from

GAUSS = MEAN + STD * (FA - 6.)

SECTION 3

SAMPLE OUTPUTS

The following sample output has been computed for the default values in the initialization files.

3-1 Initialization

These pages of printout list each of the module initialization data used in this run.

```
6/23/78 SAMPLE RUN WITH DEFAULT DATA

PHYSICAL DATA FILE
   WE           .72921151D-04      RAD/SEC
   RE           20925640.          FT
   G            32.200000          FT/SEC2
   FPNTDT       6.0000000          SECS


SEQUENCER   INITIALIZATION
   DT            .10000000          SEC
   PRNTSW       1.0000000
   OUTPSW        .0
   OFILE                 6
   PRNTDT       6.0000000
   MODFDT       6.0000000
   TEND         24.000000
   DTSLOW        .20000000D-01

TRAJECTORY  INITIALIZATION
   DT            .10000000          SEC
   FPNTSW       1.0000000
   OUTSW                  0
   OFILE                 6
   MODFDT       6.0000000
   PRNTDT       6.0000000


   LAT(DEG)             40.000000
   LON(DEG)              .0
   ALT(FT)             2000.0000
   VEL(FT/SEC)         1000.0000      1000.0000      100.00000


   ROLL(DEGS)         .0
   PITCH(DEGS)        10.000000
   YAW(DEGS)          90.000000
   WANDER(DEGS)       45.000000
   AB(FT/SEC2)        -5.5830645       .11350737D-04   -31.663198
   WB(RADS/SEC)        .53860307D-04   .10711585D-03    .78756353D-04


ENVIRONMENT  INITIALIZATION
   DT            .10000000          SEC
   PRNTSW       1.0000000
   OUTSW                  0
   OFILE                 6
   PRNTDT       6.0000000
   MODFDT       6.0000000
   VIBSW         .0

   VIBRATION GENERATOR PARAMETERS
                              AP            WH          WO
   VERTICAL LOAD
       4 PEAKS          .13603100D-03    3.0000000            13.500000
```

.154880000D-04	3.0000000	21.000000
.161500000D-05	6.0000000	32.300000
.175000000D-07	4.0000000	61.500000

LATERAL LOAD
3 PEAKS

.103040000D-04	3.5000000	16.000000
.590000000D-07	8.0000000	34.000000
.308000000D-08	6.0000000	68.000000

LONGITUDINAL LOAD
0 PEAKS

PITCH RATE
3 PEAKS

.262000000D-06	3.3000000	13.500000
.640885000D-07	2.0000000	21.000000
.132260000D-07	6.0000000	32.300000

YAW RATE PAD
4 PEAKS

.241902500D-06	3.0000000	3.0000000
.240015000D-07	5.0000000	15.000000
.288500000D-09	9.0000000	33.000000
.410000000D-10	3.0000000	67.800000

ROLL RATE PSD
1 PEAKS

.720000000D-04	2.0000000	2.0000000

GYROSCOPE INITIALIZATION

DT	.100000000D-01	SEC
PRNTSW	1.0000000	
MODPDT	6.0000000	
PRNTDT	6.0000000	
OUTSW	0	
OFILE	6	

QGBX =	1.0000000	.0	.0
	.0	.0	1.0000000
	.0	-1.0000000	.0
QGBY =	.0	1.0000000	.0
	1.0000000	.0	.0
	.0	.0	-1.0000000
QGBZ =	.0	.0	1.0000000
	1.0000000	.0	.0
	.0	1.0000000	.0

QUANT .700000000D-04 ARCSEC

K	.310000000D+10	GM CM**2/SEC**2
I	226.00000	GM CM**2
DELI	14.000000	GM CM**2

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
CO             7000000.0        DYNE CM/RAD/SEC
H              151000.00        GM CM**2/SEC

                    X                       Y                      Z
KI             1.1200000            1.1200000              1.1200000       DEG/HR/G
KO             -.320000000D-01      -.320000000D-01        -.320000000D-01 DEG/HR/G
KS             -.62000000           -.62000000             -.62000000      DEG/HR/G
KII            .0                   .0                     .0              DEG/HR/G**2,
KSS            -.300000000D-01      -.300000000D-01        -.300000000D-01 DEG/HR/G**2,
KIO            .0                   .0                     .0              DEG/HR/G**2,
KIS            .21000000            .21000000              .21000000       DEG/HR/G**2,
KOS            -.750000000D-02      -.750000000D-02        -.750000000D-02 DEG/HR/G**2,
BIAS           -.31000000           -.31000000             -.31000000      DEG/HR
BIASV          .0                   .0                     .0              (DEG/HR)**2
SFPO  ,        -20.000000           -20.000000             -20.000000      PPM
SFMO  ,        -54.000000           -54.000000             -54.000000      PPM
SFP1  ,        .0                   .0                     .0              PPM/RAD/SEC
SFM1  ,        .0                   .0                     .0              PPM/RAD/SEC

TRANS1         .0                   .0                     .0              DEG/HR
TRANTC         200.00000            200.00000              200.00000       SEC

BIASTC         40.000000        SEC
ORDER             0

ACCELEROMETER  INITIALIZATION
  DT             .100000000D-01     SEC
  PRNTSW         1.0000000
  MODEDT         6.0000000
  PRNIDT         6.0000000
  OUTSW                      0
  OFILE                     6
  GDAX =         1.0000000          .0                     .0
               .0                  .0                    1.0000000
               .0                 -1.0000000             .0
  GDAY =         .0                 1.0000000              .0
               1.0000000           .0                     .0
               .0                  .0                    -1.0000000
  GDAZ =         .0                 .0                     1.0000000
               1.0000000           .0                     .0
               .0                  1.0000000              .0

  I              5.0000000      GM CM**2
  DELI           .0             GM CM**2
  CO             25000.000      DYNE CM/RAD/SEC
  HRC            .70000000      GM CM
  QUANT          1.0000000      CM/SEC
  K              .100000000D+09 GM CM**2/RAD SEC**2
                    X                      Y                      Z
  KO             .0                   .0                     .0              MICRO G/G
  KP             .0                   .0                     .0              MICRO G/G
  KII            .80000000            .80000000              .80000000       MICRO G/G**2
  KPP            .0                   .0                     .0              MICRO G/G**2
  KIO            .0                   .0                     .0              MICRO G/G**2
  KIP            .0                   .0                     .0              MICRO G/G**2
  KOP            .0                   .0                     .0              MICRO G/G**2
  BIAS           900.00000            900.00000              900.00000       MICRO G
  BIASV          .0                   .0                     .0              (MICRO G)**2
  SFGPO          10.000000            10.000000              10.000000       PPM
```

SFGM0	10.000000	10.000000	10.000000	PPM
SFGP1	.0	.0	.0	PPM /G
SFGM1	.0	.0	.0	PPM /G
RX	1.0000000	.0	.0	FT
RY	.90000000	.20000000	.0	FT
RZ	.90000000	.0	.20000000	FT
ORDER	0			

ALTIMETER INITIALIZATION

DT	.10000000D-01	SEC
PRNTSW	1.0000000	
OUTSW	0	
OFILE	6	
PRNTDT	6.0000000	
MODPDT	6.0000000	
NOISSW	0	

ALTIMETER UNCERTAINTIES

TC	40.000000
U0(FT-2)	.39999999E-13
U1(SEC4/FT2)	.22999998E-07
U2(FT2)	17000.000
U3	.62499983E-09
U4(FT2)	97.000000

READER INITIALIZATION

DT	.99999979E-02	SEC
PRNTSW	1.0000000	
OUTSW	.0	
OFILE	6	
MODPDT	6.0000000	
PRNTDT	6.0000000	

ACC COMPENSATION INITIALIZATION

DT	.10000000D-01	SEC
PRNTSW	1.0000000	
OUTSW	0	
OFILE	6	
SPARE	.0	
MODPDT	6.0000000	
PRNTDT	6.0000000	

DELI	.0	GM CM**2		
	X	Y	Z	
BIAS	890.00000	890.00000	890.00000	MICRO G
KII	.40000000	.40000000	.40000000	MICRO G/G**2
MRC	.70000000	GM CM		
IXX	5.0000000	GM CM**2		
QADX	1.0000000	.0	.0	
	.0	.0	1.0000000	
	.0	-1.0000000	.0	
QABY	.0	1.0000000	.0	
	1.0000000	.0	.0	
	.0	.0	-1.0000000	
QADZ	.0	.0	1.0000000	

```
             1.0000000          .0                 .0
              .0               1.0000000           .0

   GMIS      1.0000000          .0                 .0
              .0               1.0000000           .0
              .0                .0                1.0000000
   SFMO      10.000000         10.000000          10.000000        PPM
   SFM1       .0                .0                 .0              PPM /G
   SFPO      10.000000         10.000000          10.000000        PPM
   SFP1       .0                .0                 .0              PPM /G
   RX        1.0100000          .0                 .0              FT
   RY         .89000000         .19000000          .0              FT
   RZ         .91000000         .0                 .19000000       FT

GYR COMPENSATION INITIALIZATION
   DT         .10000000D-01     SEC
   PRNTSW    1.0000000
   OUTSW                  0
   OFILE                 6
   MODPDT    6.0000000
   PRNTDT    6.0000000


              X                 Y                  Z
   BIAS      -.34000000        -.34000000         -.34000000       DEG/HR
   SFPO      -30.000000        -30.000000         -30.000000       PPM
   SFMO      -44.000000        -44.000000         -44.000000       PPM

   GCDX      1.0000000          .0                 .0
              .0                .0                1.0000000
              .0               -1.0000000          .0

   GCDY       .0               1.0000000           .0
             1.0000000          .0                 .0
              .0                .0                -1.0000000

   GCDZ       .0                .0                1.0000000
             1.0000000          .0                 .0
              .0               1.0000000           .0
   GMIS      1.0000000          .0                 .0
              .0               1.0000000           .0
              .0                .0                1.0000000


   KI        1.1600000         1.1600000          1.1600000        DEG/HR/G
   KO        -.37000000D-01    -.37000000D-01     -.37000000D-01   DEG/HR/G
   KS        -.66000000        -.66000000         -.66000000       DEG/HR/G
   KII        .0                .0                 .0              DEG/HR/G**2,
   KIS        .60000000D-01     .60000000D-01      .60000000D-01   DEG/HR/G**2,
   KOS       -.15000000D-01    -.15000000D-01     -.15000000D-01   DEG/HR/G**2,
   KSS        .0                .0                 .0              DEG/HR/G**2,
   KIO        .0                .0                 .0              DEG/HR/G**2,
   H         151000.00      GM CM**2/SEC
   DELI      14.000000      GM CM**2
   IXX       226.00000      GM CM**2

   SFM1       .0                .0                 .0              PPM /RAD/SEC
   SFP1       .0                .0                 .0              PPM /RAD/SEC
```

```
QIB=          0.67636424    -0.70710678     -0.12278780
              0.64506455     0.54167522      0.53896131
             -0.31459208    -0.45451948      0.83333299
ALGORITHM INITIALIZATION
   DT              .10000000          SEC
   PRNTSW          1.0000000
   OUTSW                       0
   OFILE                       6
   CORDCM          1.0000000
   MODPDT          6.0000000
   PRNTDT          6.0000000
   QUAT         .0                .0              .0             .0
   ORDER           3
   DTNRM           4.0000000
   LATERR          .0            DEG
   LONERR          .0            DEG
   HANERR          .0            DEG
   PITERR          .0            DEG
   ROLERR          .0            DEG
   YAWERR          .0            DEG

                A TRANSPOSE
     .76604444        .0             .64278761
    -.45451948        .70710678      .54167522
    -.45451948       -.70710678      .54167522

NAVIGATION   INITIALIZATION
   DT              .10000000          SEC
   PRNTSW          1.0000000
   OUTSW           .0
   OFILE                       6
   SPARE           .0
   PRNTDT          6.0000000
   CVD1(SEC-1)          .600000000D-01
   CVD2(SEC-2)          .162000000D-02
   CVD3(SEC-3)          .162000000D-04


   INITIAL VEHICLE POSITION
      H(FT)           2000.0000
      ALTERR(FT)      .0
      V(FT/SEC)       1000.0000         1000.0000        100.00000
      VERR(FT/SEC)       .0                .0               .0
      LAT(DEGS)       40.000000
      LATERR(DEGS)     .0
      LON(DEGS)       .0
      LONERR(DEGS)     .0

EVALUATION INITIALIZATION
   DT              .10000000          SEC
   PRNTSW          1.0000000
   OUTSW                       0
   OFILE                       6
   PRNTDT          1.0000000
```

3.2 MODULE

This sample page printed out every PRNTDT or MODPDT interval display the values of significant variables at the printing time for each of the modules.

```
6/23/78 SAMPLE RUN WITH DEFAULT DATA

    ** SEQ **  TIME=    .10000000

     ** TRJ **    AB(FT/SEC2)     5.5614927       .18130318        31.456426
                  WB(RAD/SEC)     .538607850-04   .107115590-03    .707570280-04
                  LAT(DEGS)       40.000274
                  LON(DEGS)       .356901650-03
                  ALF(DEGS)       44.999771
                  ALT(FT)         2010.0000
                  VEL(FT/SEC)     1000.0000       1000.0000        100.00000
                  DV(FT/SEC)      -.117841500-01  .138559920-01    3.1944275

                  QBI      .69635999      .64506335      -.31460459
                          -.70711293      .54167304      -.45451251
                          -.12277654      .53896530       .83333206

     ** ENV **   AB(FT/SEC2)      5.5614927       .18130318        31.456426
                 WB(RAD/SEC)      .538607850-04   .107115590-03    .787570280-04
                 AUB(FT/SEC2)     5.5614927       .18130318        31.456426
                 WEB(RAD/SEC)     .538607850-04   .107115590-03    .787570280-04
                 WBDOT(R/S2)      .478246370-08   -.255999640-08   .674477150-08

     ** GYR **   DTHETA(RADS)     .545604470-05   .105015430-04    .749904950-05

     ** ACC **   DV(FT/SEC)       .55278749       .150564690-01    3.1425001

     ** ALT **   ALT(FT)          2010.0000

     ** RDR **   DV(FT/SEC)       .55278749       .150564690-01    3.1425001
                 DTHETA(RADS)     .545604470-05   .105015430-04    .749904950-05

     ** CAC **   DV (FT/SEC)      .55565860       .179224120-01    3.1453971

     ** CGY **   DTHETA (RAD)       .537233000-05   .107301130-04    .787876770-05

     ** ALG **   DVH(FT/SEC)      2.4382064       -.119321910-01   2.0633971

                   DCM   -.31459953      -.45451768       .83333116
                          .69636220      -.70710961      -.12278261
                          .64506334       .54167303       .53096532

     ** LLN **   H(FT)            2010.0000
                 V(FT/SEC)        999.99904       999.99950        99.999671
                 LAT(DEGS)        40.000274
                 LON(DEGS)        .356901000-03
               PITCH(DEGS)        9.9999989
                ROLL(DEGS)        -.816306260-06
             HEADING(DEGS)        45.000000

           A TRANSPOSE
     .76604136       .477174630-05    .64279128
    -.45452466       .70710678        .54167087
    -.45451947      -.70710678        .54167521

                 DTEM
    .17364016       -.140307850-07    .98480776
    .98480776       -.400495900-05   -.17364016
    .394615710-05    1.0000000       -.681566670-06
```

3.3 EVALUATION

Two sample pages display the 50 module cycle computations of trajectory parameters and navigational errors.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

TIME (SEC)	LAT (DEG)	LON (DEG)	TRAJECTORY ALT (FT)	VX(EAST) (FPS)	VY(NORTH) (FPS)	VZ(UP) (FPS)	HEAD (DEG)	PITCH (DEG)	ROLL (DEG)
0.100	40.000	.356900D-03	2010.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.200	40.001	.713800D-03	2020.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.300	40.001	.107070D-02	2030.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.400	40.001	.142750D-02	2040.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.500	40.001	.178450D-02	2050.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.600	40.002	.214140D-02	2060.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.700	40.002	.249830D-02	2070.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.800	40.002	.285520D-02	2080.0	1000.0	1000.0	100.00	45.000	10.000	.0
0.900	40.002	.321220D-02	2090.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.000	40.003	.356910D-02	2100.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.100	40.003	.392600D-02	2110.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.200	40.003	.428290D-02	2120.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.300	40.004	.463970D-02	2130.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.400	40.004	.499670D-02	2140.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.500	40.004	.535370D-02	2150.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.600	40.004	.571060D-02	2160.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.700	40.005	.606750D-02	2170.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.800	40.005	.642440D-02	2180.0	1000.0	1000.0	100.00	45.000	10.000	.0
1.900	40.005	.678130D-02	2190.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.000	40.005	.713820D-02	2200.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.100	40.006	.749520D-02	2210.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.200	40.006	.785210D-02	2220.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.300	40.006	.820910D-02	2230.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.400	40.007	.856600D-02	2240.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.500	40.007	.892290D-02	2250.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.600	40.007	.927980D-02	2260.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.700	40.007	.963680D-02	2270.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.800	40.008	.999370D-02	2280.0	1000.0	1000.0	100.00	45.000	10.000	.0
2.900	40.008	.103510D-01	2290.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.000	40.008	.107080D-01	2300.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.100	40.009	.110650D-01	2310.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.200	40.009	.114210D-01	2320.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.300	40.009	.117780D-01	2330.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.400	40.009	.121350D-01	2340.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.500	40.010	.124920D-01	2350.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.600	40.010	.128490D-01	2360.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.700	40.010	.132060D-01	2370.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.800	40.010	.135630D-01	2380.0	1000.0	1000.0	100.00	45.000	10.000	.0
3.900	40.011	.139200D-01	2390.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.000	40.011	.142770D-01	2400.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.100	40.011	.146340D-01	2410.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.200	40.012	.149910D-01	2420.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.300	40.012	.153480D-01	2430.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.400	40.012	.157050D-01	2440.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.500	40.012	.160620D-01	2450.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.600	40.013	.164190D-01	2460.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.700	40.013	.167760D-01	2470.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.800	40.013	.171330D-01	2480.0	1000.0	1000.0	100.00	45.000	10.000	.0
4.900	40.013	.174900D-01	2490.0	1000.0	1000.0	100.00	45.000	10.000	.0
5.000	40.014	.178470D-01	2500.0	1000.0	1000.0	100.00	45.000	10.000	.0

NAVIGATION ERRORS

TIME (SEC)	LAT (FT)	LON (FT)	ALT (FT)	VX(EAST) (FPS)	VY(NORTH) (FPS)	VZ(UP) (FPS)	HEAD (SEC)	PITCH (SEC)	ROLL (SEC)
0.100	.73791D-03	.18397D-03	.16428D-04	.16450D-03	.45858D-03	.32855D-03	.63522D-03	.38250D-02	.29387D-02
0.200	[illegible]	.36626D-03	.64096D-04	.32887D-03	.91715D-03	.65415D-03	.14125D-02	.77238D-02	.58271D-02
0.300	.84202D-03	.56757D-03	.13484D-03	-.25579D-03	.62681D-03	.79004D-03	.21020D-02	.11624D-01	.87339D-02
0.400	.93522D-03	.72729D-03	.22837D-03	-.91634D-04	.10554D-02	.11097D-02	.27790D-02	.15460D-01	.11606D-01
0.500	.97707D-03	.86721D-03	.34290D-03	-.67651D-03	.79508D-03	.12396D-02	.35137D-02	.19370D-01	.14447D-01
0.600	.10556D-02	.99211D-03	.42667D-03	-.33052D-03	.13857D-02	.49422D-03	.42720D-02	.23303D-01	.17402D-01
0.700	.11677D-02	.11779D-02	.48870D-03	.54301D-03	.10839D-02	.80501D-03	.49603D-02	.27190D-01	.20251D-01
0.800	.12402D-02	.13307D-02	.57086D-03	-.41306D-04	.79350D-03	.92606D-03	.56518D-02	.31040D-01	.23150D-01
0.900	.13211D-02	.15624D-02	.67431D-03	.12223D-03	.12522D-02	.12309D-02	.64031D-02	.34966D-01	.26031D-01
1.000	.14105D-02	.17230D-02	.79729D-03	-.46318D-03	.96193D-03	.13460D-02	.71383D-02	.38825D-01	.28933D-01
1.100	.15149D-02	.18592D-02	.88302D-03	-.16772D-03	.15526D-02	.56570D-03	.78301D-02	.42711D-01	.31803D-01
1.200	.16344D-02	.20009D-02	.94519D-03	-.75332D-03	.17624D-02	.69488D-03	.85375D-02	.46649D-01	.34709D-01
1.300	.17242D-02	.21495D-02	.10238D-02	.17035D-03	.96058D-03	.98780D-03	.92844D-02	.50557D-01	.37561D-01
1.400	.18219D-02	.23523D-02	.11290D-02	.33344D-03	.14192D-02	.12778D-02	.99651D-02	.54432D-01	.40495D-01
1.500	.19280D-02	.25340D-02	.12544D-02	-.25250D-03	.11290D-02	.13781D-02	.10705D-01	.58313D-01	.43327D-01
1.600	.20425D-02	.26946D-02	.13976D-02	-.89633D-04	.15877D-02	.16621D-02	.11428D-01	.62196D-01	.46266D-01
1.700	.21655D-02	.23341D-02	.15583D-02	-.67579D-03	.12975D-02	.17564D-02	.12140D-01	.66085D-01	.49093D-01
1.800	.23036D-02	.29598D-02	.16845D-02	-.38108D-03	.18882D-02	.97530D-03	.12876D-01	.69997D-01	.52033D-01
1.900	.24136D-02	.31073D-02	.17749D-02	-.20696D-03	.83756D-03	.10636D-02	.13540D-01	.73661D-01	.54829D-01
2.000	.25939D-02	.32724D-02	[illegible]	-.44522D-04	.12963D-02	.13358D-02	.14279D-01	.77759D-01	.57765D-01
2.100	.26352D-02	.34530D-02	.20184D-02	.11752D-03	.17550D-02	.16049D-02	.15007D-01	.81663D-01	.60700D-01
2.200	.27750D-02	.36139D-02	.21697D-02	-.46887D-03	.14648D-02	.16844D-02	.15721D-01	.85503D-01	.63523D-01
2.300	.29232D-02	.37529D-02	.23367D-02	-.30675D-03	.19236D-02	.19475D-02	.16419D-01	.89413D-01	.66451D-01
2.400	.30795D-02	.38707D-02	.25204D-02	-.89365D-03	.16334D-02	.20210D-02	.17105D-01	.93259D-01	.69337D-01
2.500	.32135D-02	.40118D-02	.26663D-02	.16078D-03	.14638D-02	.12191D-02	.17868D-01	.97239D-01	.72183D-01
2.600	.33616D-02	.42136D-02	.27834D-02	.32260D-03	.19225D-02	.14734D-02	.18597D-01	.10109	.75113D-01
2.700	.35181D-02	.43942D-02	.29164D-02	-.26462D-03	.16324D-02	.15379D-02	.19312D-01	.10502	.78001D-01
2.800	.36331D-02	.45535D-02	.30635D-02	-.10303D-03	.20912D-02	.17862D-02	.20015D-01	.10888	.80856D-01
2.900	.38566D-02	.46916D-02	.32260D-02	-.69046D-03	.18011D-02	.18443D-02	.20702D-01	.11275	.83738D-01
3.000	.40385D-02	.48084D-02	.34021D-02	-.52909D-03	.22599D-02	.20871D-02	.21447D-01	.11662	.86667D-01
3.100	.42283D-02	.49793D-02	.36008D-02	.39266D-03	.19582D-02	.23265D-02	.22160D-01	.12060	.89498D-01
3.200	.43864D-02	.51668D-02	.38125D-02	-.19510D-03	.16681D-02	.23761D-02	.22876D-01	.12448	.92381D-01
3.300	.45636D-02	.53397D-02	.39839D-02	.97997D-04	.22590D-02	.15503D-02	.23619D-01	.12838	.95309D-01
3.400	.47539D-02	.54978D-02	.41163D-02	-.48995D-03	.19689D-02	.15940D-02	.24310D-01	.13227	.98187D-01
3.500	.49527D-02	.56345D-02	.42607D-02	-.32911D-03	.24278D-02	.18214D-02	.25058D-01	.13609	.10104
3.600	.51590D-02	.57501D-02	.44183D-02	-.91728D-03	.21377D-02	.18591D-02	.25719D-01	.13998	.10391
3.700	.53374D-02	.59022D-02	.45875D-02	.39219D-05	.16361D-02	.20805D-02	.26419D-01	.14392	.10681
3.800	.55226D-02	.60683D-02	.47772D-02	.16434D-03	.22950D-02	.22990D-02	.27125D-01	.14782	.10972
3.900	.57167D-02	.62338D-02	.49777D-02	-.42417D-03	.20049D-02	.23278D-02	.27889D-01	.15173	.11261
4.000	.59190D-02	.63764D-02	.51904D-02	-.26306D-03	.24638D-02	.25403D-02	.28567D-01	.15557	.11552
4.100	.61299D-02	.64985D-02	.54133D-02	-.85253D-03	.21738D-02	.25638D-02	.29303D-01	.15948	.11840
4.200	.63491D-02	.65959D-02	.56462D-02	-.69249D-03	.26327D-02	.27695D-02	.30026D-01	.16333	.12124
4.300	.65829D-02	.67600D-02	.58453D-02	.36002D-03	.24632D-02	.19139D-02	.30753D-01	.16731	.12413
4.400	.67937D-02	.69442D-02	.60007D-02	-.22900D-03	.21732D-02	.19277D-02	.31447D-01	.17117	.12700
4.500	.70129D-02	.71071D-02	.61568D-02	-.69225D-04	.26322D-02	.21253D-02	.32199D-01	.17503	.12992
4.600	.72406D-02	.72469D-02	.63416D-02	-.65847D-03	.23422D-02	.21332D-02	.32866D-01	.17897	.13278
4.700	.74767D-02	.73684D-02	.65245D-02	-.49391D-03	.28011D-02	.23247D-02	.33590D-01	.18283	.13569
4.800	.77589D-02	.75842D-02	.67258D-02	-.33944D-03	.32601D-02	.25133D-02	.34301D-01	.18671	.13860
4.900	.80114D-02	.76566D-02	.69346D-02	-.16853D-03	.22097D-02	.25122D-02	.35049D-01	.19062	.14145
5.000	.82342D-02	.78254D-02	.71525D-02	-.92977D-05	.26686D-02	.26949D-02	.35732D-01	.19450	.14435

R-1136

VOLUME I, II, III, IV

DISTRIBUTION LIST

A. Ciccolo
G. Coate
R. Crisp
K. Daly
M. Dare
W. Delaney
W. Denhard
J. DiSorbo
J. Feldman
K. Fertig
J. Fish
J. Gilmore
J. Goode
J. Harper
R. Harris
J. Hursh
P. Kampion
A. Kosol
W. Koenigsberg
R. Leger
J. Lombardo
D. Millard
B. McCoy
P. Motyka
R. Nurse (19)

J. Prohaska
T. Reed
D. Riegsecker
J.P. Ryan
G. Schmidt
J. Sciegienny
L. Schnee
R. Setterlund
T. Thorvaldsen
L.W. Torrey
K. Vincent
P. Volante

TIC (5)

Air Force Avionics Laboratory (30)
Attn: RWA-3 D. Kaiser
Wright-Patterson Air Force Base
Dayton, Ohio 45433

Air Force Avionics Laboratory (5)
Attn: N. Banke RWA-2
Wright-Patterson Air Force Base
Dayton, Ohio 45433